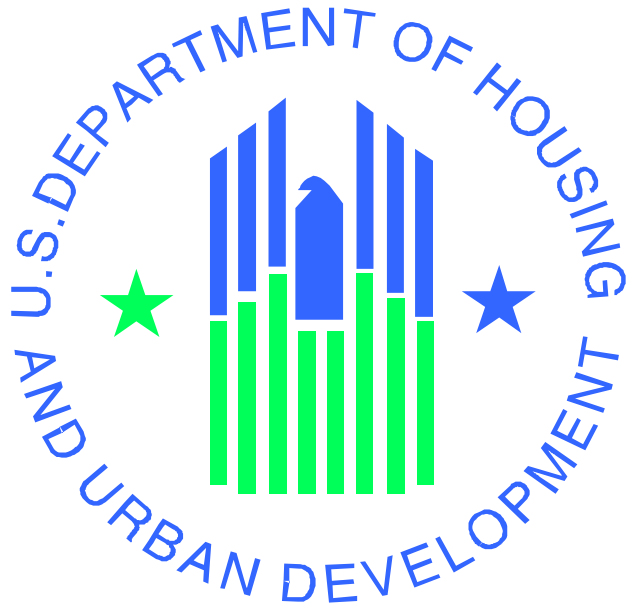
Department of Housing and Urban Development
Public and Indian Housing – Real Estate Assessment Center
(PIH-REAC)



SDM Design Phase Deliverable
Program Specifications

For The

Financial Assessment Subsystem – Public Housing
(FASS PH)

System:	REACS
Subsystem:	FASS-PH
Component:	PH
Release:	8.1.0.0
Database Release:	REACS
Doc Type:	SDM Design Phase Deliverable – Program Specifications
Doc Date:	8/26/2005
Doc Author:	Avineon
Doc Number:	1.4
Doc Status:	Final

PROGRAM SPECIFICATIONS

Financial Assessment Subsystem – Public Housing (FASS-PH)
Release 8.1.0.0

U.S. Department of Housing and Urban Development

August 26, 2005

Revision Sheet

Revision No.	Date	Revision Description
Rev. 1.0	6/14/05	Initial version.
Rev. 1.1	6/21/05	Incorporated internal comments.
Rev. 1.2	6/22/05	Incorporated Peer Review comments.
Rev. 1.3	6/30/05	Incorporated IT Manager's Comments.

Program Specifications Authorization Memorandum

I have carefully assessed the Program Specifications for the Financial Assessment Subsystem-Public Housing (FASS PH) Release 8.1.0.0. This document has been completed in accordance with the requirements of the HUD System Development Methodology.

MANAGEMENT CERTIFICATION - Please check the appropriate statement.

_____ The document is accepted.

_____ The document is accepted pending the changes noted.

_____ The document is not accepted.

We fully accept the changes as needed improvements and authorize initiation of work to proceed. Based on our authority and judgment, the continued operation of this system is authorized.

_____ _____
Freddie Harrison DATE
FASS-PH IT Manager

_____ _____
Nick Miele DATE
FASS-PH Business Program Manager

PROGRAM SPECIFICATIONS

TABLE OF CONTENTS

Page #

1.0 GENERAL INFORMATION 1-2

1.1 Purpose 1-2

1.2 Scope 1-2

1.3 System Overview 1-4

1.4 Project References 1-6

1.5 Terms and Abbreviations 1-8

1.6 Points of Contact 1-12
1.6.1 Information 1-12
1.6.2 Coordination 1-12

1.7 Master List of Programs 1-13
1.7.1 DCF Financial Statement Line Item G3000-010 1-13
1.7.2 View of Prior Years Submission Comments 1-13
1.7.3 Additional Valid Value for DCF Line Item 4200-050 1-14
1.7.4 Line Item 1102 1-14
1.7.5 Analyst Column Display of Analyst and Manager's Names 1-14
1.7.6 FDS Report Page Print Functionality 1-14
1.7.7 Review Submission Page Text Change 1-14

1.8 Infrastructure Changes 1-14
1.8.1 File Attachments Storage 1-14
1.8.2 Application Use of Port 443 1-15
1.8.3 Removal of Check Box for Guest Users 1-15
1.8.4 Identity Type 1-15
1.8.5 LOCCS/HUDCAPS Interface Change 1-15

2.0 DCF Financial statement line item G3000-010 2-17

2.1 Program Description 2-17
2.1.1 Software Unit Description 2-17
2.1.2 Software Unit 2-17
2.1.3 Accuracy and Validity 2-18
2.1.4 Timing 2-18
2.1.5 Adaptability 2-19

2.2 Environment 2-19
2.2.1 Support Software Environment 2-19
2.2.2 Interfaces 2-19
2.2.3 Storage 2-19
2.2.3.1 Internal Storage 2-20

2.2.3.2 Device Storage 2-20
2.2.3.3 Offline Storage 2-20
2.2.3.4 Temporary and Permanent Storage 2-20
2.2.4 Security 2-20
2.2.5 Communications Environment 2-20

2.3 Design Details 2-21
2.3.1 Input 2-23
2.3.1.1 Input Records 2-23
2.3.1.2 Input Data Elements 2-23
2.3.2 Output 2-23
2.3.2.1 Output Reports 2-23
2.3.2.2 Output Data Element 2-23
2.3.3 Software Relationships 2-23
2.3.4 Software Unit Logic 2-23

3.0 PRIOR YEARS SUBMISSION COMMENTS REVIEW 2-27

3.0 Prior years submission comments review 3-28

3.1 Program Description 3-28
3.1.1 Software Unit Description 3-28
3.1.2 Software Unit 3-29
3.1.3 Accuracy and Validity 3-29
3.1.4 Timing 3-29
3.1.5 Adaptability 3-30

3.2 Environment 3-30
3.2.1 Support Software Environment 3-30
3.2.2 Interfaces 3-31
3.2.3 Storage 3-31
3.2.3.1 Internal Storage 3-31
3.2.3.2 Device Storage 3-31
3.2.3.3 Offline Storage 3-31
3.2.3.4 Temporary and Permanent Storage 3-31
3.2.4 Security 3-31
3.2.5 Communications Environment 3-32

3.3 Design Details 3-33
3.3.1 Input 3-35
3.3.1.1 Input Records 3-35
3.3.1.2 Input Data Elements 3-35
3.3.2 Output 3-35
3.3.2.1 Output Reports 3-35
3.3.2.2 Output Data Elements 3-35
3.3.3 Software Relationships 3-35
3.3.4 Software Unit Logic 3-37

4.0 ADDITIONAL VALID VALUE FOR DCF LINE ITEM 4200-050 3-47

4.0 ADDITIONAL VALID VALUE FOR DCF LINE ITEM 4200-050 4-48

4.1 Program Description 4-48

4.1.1 Software Unit Description 4-48
4.1.2 Software Unit 4-48
4.1.3 Accuracy and Validity 4-48
4.1.4 Timing 4-49
4.2.6 Adaptability 4-49

4.2 Environment 4-49
4.2.1 Support Software Environment 4-49
4.2.2 Interfaces 4-50
4.2.3 Storage 4-50
4.2.3.1 Internal Storage 4-50
4.2.3.2 Device Storage 4-50
4.2.3.3 Offline Storage 4-51
4.2.3.4 Temporary and Permanent Storage 4-51
4.2.4 Security 4-51
4.2.5 Communications Environment 4-51

4.3 Design Details 4-52
4.3.1 Input 4-54
4.3.1.1 Input Records 4-54
4.3.1.2 Input Data Elements 4-54
4.3.2 Output 4-54
4.3.2.1 Output Reports 4-54
4.3.2.2 Output Data Elements 4-54
4.3.3 Software Relationships 4-54
4.3.4 Software Unit Logic 4-54

5.0 LINE ITEM 1102 4-57

5.0 Line Item 1102 5-58

5.1 Program Description 5-58
5.1.1 Software Unit Description 5-58
5.1.2 Software Unit 5-58
5.1.3 Accuracy and Validity 5-58
5.1.4 Timing 5-58
5.1.5 Adaptability 5-59

5.2 Environment 5-59
5.2.1 Support Software Environment 5-59
5.2.2 Interfaces 5-60
5.2.3 Storage 5-60
5.2.3.1 Internal Storage 5-60
5.2.3.2 Device Storage 5-60
5.2.3.3 Offline Storage 5-60
5.2.3.4 Temporary and Permanent Storage 5-61
5.2.4 Security 5-61
5.2.5 Communications Environment 5-61

5.3 Design Details 5-62
5.3.1 Input 5-65
5.3.1.1 Input Records 5-65

5.3.1.2 Input Data Elements 5-65
5.3.2 Output 5-65
5.3.2.1 Output Reports 5-65
5.3.2.2 Output Data Elements 5-65
5.3.3 Software Relationships 5-65
5.3.4 Software Unit Logic 5-66

6.0 FASS-PH ANALYST INBOX 5-68

6.0 FASS Analyst inbox 6-69

6.1 Program Description 6-69
6.1.1 Software Unit Description 6-69
6.1.2 Software Unit 6-69
6.1.3 Accuracy and Validity 6-69
6.1.4 Timing 6-70
6.1.5 Adaptability 6-70

6.2 Environment 6-71
6.2.1 Support Software Environment 6-71
6.2.2 Interfaces 6-71
6.2.3 Storage 6-71
6.2.3.1 Internal Storage 6-71
6.2.3.2 Device Storage 6-72
6.2.3.3 Offline Storage 6-72
6.2.3.4 Temporary and Permanent Storage 6-72
6.2.4 Security 6-72
6.2.5 Communications Environment 6-72

6.3 Design Details 6-73
6.3.1 Input 6-74
6.3.1.1 Input Records 6-74
6.3.1.2 Input Data Elements 6-74
6.3.2 Output 6-74
6.3.2.1 Output Reports 6-73
6.3.2.2 Output Data Elements 6-74
6.3.3 Software Relationships 6-74
6.3.4 Software Unit Logic 6-75

7.0 FDS REPORT PAGE PRINT FUNCTIONALITY 6-78

7.0 FDS REPORT PAGE PRINT FUNCTIONALITY 7-79

7.1 Program Description 7-79
7.1.1 Software Unit Description 7-79
7.1.2 Software Unit 7-79
7.1.3 Accuracy and Validity 7-79
7.1.4 Timing 7-79
7.1.5 Adaptability 7-80

7.2 Environment 7-80
7.2.1 Support Software Environment 7-80
7.2.2 Interfaces 7-81

7.2.3 Storage 7-81
7.2.3.1 Internal Storage 7-81
7.2.3.2 Device Storage 7-81
7.2.3.3 Offline Storage 7-81
7.2.3.4 Temporary and Permanent Storage 7-82
7.2.4 Security 7-82
7.2.5 Communications Environment 7-82

7.3 Design Details 7-83
7.3.1 Input 7-84
7.3.1.1 Input Records 7-84
7.3.1.2 Input Data Elements 7-84
7.3.2 Output 7-84
7.3.2.1 Output Reports 7-83
7.3.2.2 Output Data Elements 7-84
7.3.3 Software Relationships 7-84
7.3.4 Software Unit Logic 7-85

8.0 FILE ATTACHMENTS STORAGE 7-87

8.0 FILE ATTACHMENTS STORAGE 8-88

8.1 Program Description 8-88
8.1.1 Software Unit Description 8-88
8.1.2 Software Unit 8-88
8.1.3 Accuracy and Validity 8-88
8.1.4 Timing 8-89
8.1.5 Adaptability 8-89

8.2 Environment 8-90
8.2.1 Support Software Environment 8-90
8.2.2 Interfaces 8-90
8.2.3 Storage 8-90
8.2.3.1 Internal Storage 8-90
8.2.3.2 Device Storage 8-91
8.2.3.3 Offline Storage 8-91
8.2.3.4 Temporary and Permanent Storage 8-91
8.2.4 Security 8-91
8.2.5 Communications Environment 8-91

8.3 Design Details 8-91
8.3.1 Input 8-92
8.3.1.1 Input Records 8-92
8.3.1.2 Input Data Elements 8-92
8.3.2 Output 8-92
8.3.2.1 Output Reports 8-91
8.3.2.2 Output Data Elements 8-92
8.3.3 Software Relationships 8-92
8.3.4 Software Unit Logic 8-93

9.0 RELATIVE SERVER NAME 8-103

9.0 REALTIVE SERVER NAME 9-104

9.1 Program Description 9-104
9.1.1 Software Unit Description 9-104
9.1.2 Software Unit 9-104
9.1.3 Accuracy and Validity 9-104
9.1.4 Timing 9-104
9.1.5 Adaptability 9-105

9.2 Environment 9-105
9.2.1 Support Software Environment 9-105
9.2.2 Interfaces 9-106
9.2.3 Storage 9-106
9.2.3.1 Internal Storage 9-106
9.2.3.2 Device Storage 9-106
9.2.3.3 Offline Storage 9-107
9.2.3.4 Temporary and Permanent Storage 9-107
9.2.4 Security 9-107
9.2.5 Communications Environment 9-107

9.3 Design Details 9-107
9.3.1 Input 9-108
9.3.1.1 Input Records 9-108
9.3.1.2 Input Data Elements 9-108
9.3.2 Output 9-108
9.3.2.1 Output Reports 9-107
9.3.2.2 Output Data Elements 9-108
9.3.3 Software Relationships 9-108
9.3.4 Software Unit Logic 9-109

10.0 CHECK BOX FOR GUEST USERS 9-111

10.0 CHECK BOX FOR GUEST USERS 10-112

10.1 Program Description 10-112
10.1.1 Software Unit Description 10-112
10.1.2 Software Unit 10-112
10.1.3 Accuracy and Validity 10-113
10.1.4 Timing 10-113
10.1.4.1 Adaptability 10-114

10.2 Environment 10-114
10.2.1 Support Software Environment 10-114
10.2.2 Interfaces 10-114
10.2.3 Storage 10-114
10.2.3.1 Internal Storage 10-115
10.2.3.2 Device Storage 10-115
10.2.3.3 Offline Storage 10-115
10.2.3.4 Temporary and Permanent Storage 10-115
10.2.4 Security 10-115
10.2.5 Communications Environment 10-115

10.3 Design Details 10-116
10.3.1 Input 10-116

10.3.1.1 Input Records 10-117
10.3.1.2 Input Data Elements 10-117
10.3.2 Output 10-117
10.3.2.1 Output Reports 10-117
10.3.2.2 Output Data Elements 10-117
10.3.3 Software Relationships 10-117
10.3.4 Software Unit Logic 10-118

11.0 IDENTITY TYPE 10-120

11.0 IDENTITY TYPE 11-121

11.1 Program Description 11-121
11.1.1 Software Unit Description 11-121
11.1.2 Software Unit 11-121
11.1.3 Accuracy and Validity 11-121
11.1.4 Timing 11-122
11.1.5 Adaptability 11-122

11.2 Environment 11-123
11.2.1 Support Software Environment 11-123
11.2.2 Interfaces 11-123
11.2.3 Storage 11-123
11.2.3.1 Internal Storage 11-124
11.2.3.2 Device Storage 11-124
11.2.3.3 Offline Storage 11-124
11.2.3.4 Temporary and Permanent Storage 11-124
11.2.4 Security 11-124
11.2.5 Communications Environment 11-124

11.3 Design Details 11-125
11.3.1 Input 11-127
11.3.1.1 Input Records 11-127
11.3.1.2 Input Data Elements 11-127
11.3.2 Output 11-127
11.3.2.1 Output Reports 11-125
11.3.2.2 Output Data Elements 11-127
11.3.3 Software Relationships 11-128
11.3.4 Software Unit Logic 11-129

12.0 LOCCS/HUDCAPS INTERFACE CHANGE 11-146

12.0 LOCCS/HUDCAPS INTERFACE CHANGE 12-147

12.1 Program Description 12-147
12.1.1 Software Unit Description 12-147
12.1.2 Software Unit 12-147
12.1.3 Accuracy and Validity 12-147
12.1.4 Timing 12-148
12.1.5 Adaptability 12-148

12.2 Environment 12-148
12.2.1 Support Software Environment 12-149

12.2.2 Interfaces 12-149
12.2.3 Storage 12-149
12.2.3.1 Internal Storage 12-149
12.2.3.2 Device Storage 12-149
12.2.3.3 Offline Storage 12-150
12.2.3.4 Temporary and Permanent Storage 12-150
12.2.4 Security 12-150
12.2.5 Communications Environment 12-150

12.3 Design Details 12-151
12.3.1 Input 12-152
12.3.1.1 Input Records 12-152
12.3.1.2 Input Data Elements 12-152
12.3.2 Output 12-152
12.3.2.1 Output Reports 12-150
12.3.2.2 Output Data Elements 12-152
12.3.3 Software Relationships 12-152
12.3.4 Software Unit Logic 12-153

13.0 Review Submission Page Text Change 13-173

13.0 Review Submission Page Text Change 13-173

13.1 Program Description 13-173
13.1.1 Software Unit Description 13-173
13.1.2 Software Unit 13-173
13.1.3 Accuracy and Validity 13-172
13.1.4 Timing 13-173
13.1.5 Adaptability 13-175

13.2 Environment 13-175
13.2.1 Support Software Environment 13-175
13.2.2 Interfaces 13-175
13.2.3 Storage 13-176
13.2.3.1 Internal Storage 13-176
13.2.3.2 Device Storage 13-176
13.2.3.3 Offline Storage 13-176
13.2.3.4 Temporary and Permanent Storage 13-176
13.2.4 Security 13-176
13.2.5 Communications Environment 13-176

13.3 Design Details 13-177
13.3.1 Input 13-178
13.3.1.1 Input Records 13-178
13.3.1.2 Input Data Elements 13-178
13.3.2 Output 13-178
13.3.2.1 Output Reports 13-178
13.3.2.2 Output Data Elements 13-178
13.3.3 Software Relationships 13-178
13.3.4 Software Unit Logic 13-178

APPENDIX A: BUSINESS RULES 13-180

1.0 GENERAL INFORMATION

1.0 GENERAL INFORMATION

1.1 Purpose

The Department of Housing and Urban Development (HUD) has established the HUD Annual Performance Plan (APP) to demonstrate how HUD is measuring progress toward achieving its critical mission: to promote adequate and affordable housing, economic opportunity, and a suitable living environment free from discrimination. Financial Assessment Subsystem – Public Housing (FASS-PH) aids HUD in meeting the following departmental goals and objectives: restoring the public trust, increasing affordable housing and reducing homelessness.

FASS-PH also aids HUD in addressing existing material weaknesses and business operating goals. Specifically, the FASS-PH functions address the following HUD material weaknesses identified by IG audit: Payment of Incorrect Subsidy Amount and HUD Resource Management. FASS-PH supports business operating plan goal 6.1.4 related to decreasing the share of public housing units managed by troubled housing authorities.

The FASS-PH Release 8.1.0.0 SDM Program Specifications Document is the second document in a series of two HUD System Development Methodology (SDM) Design Phase deliverable documents required from the FASS System Development Team. This series of SDM Design Phase deliverables includes the System/Subsystem Specifications and the Program Specifications. This document outlines the FASS-PH Release 8.1.0.0 Program Specifications.

1.2 Scope

FASS-PH Release 8.1.0.0 will expand FASS-PH capabilities as of Release 8.0.0.0 and is scheduled to be implemented on August 26, 2005. The following table describes the functionality to be implemented as part of FASS-PH Release 8.1.0.0.

Below is a list of the FASS-PH Release 8.1.0.0 requirements:

Requirement Table			
#	Req. #	Title	Description
1	2	Line Item G3000-010	First, the DCF/Financial Statement/G3000-010 Type of Audit Report/G3000-060 & G3000-070 will now reflect Fund Type and Opinion of the Fund rather than Program. Auditors should only be entering opinion for funds within the PHA.
2	3	View Prior Fiscal Year Submission Comments	Allow analysts to review prior years submission comments while still reviewing the current FYE submission. This will allow the analyst to review prior submission comments without navigating between multiple submissions.

Requirement Table			
#	Req. #	Title	Description
3	4	Line Item G4200-010 & G4200-050	Modify Line Item 4200-050 to default to "N/A"; if and only if Line Item 4200-010 is selected "No" for Non-Major Programs audited A133, there will be no penalty when this opinion is selected.
4	5	Line Item 1102	Modify line item 1102 so that the external user will not be able to enter data in the field. This new methodology should begin for all 9/30/2005 submissions.
5	6.2	FASS Analyst Column	Modify the FASS Analyst column for the external user inbox only to display the name of the Business Manager or Analyst.
6	6.7	FDS Report	Repair the FDS report page to print correctly from MS Internet Explorer.
7	7.1	Storing Assessment Attachments	Change the storage of permanent file attachments from being part of the UNIX /Windows file system to being stored as Binary Large Objects (BLOBs) in the database. All file attachments need to be stored and retrieved on the REACS database.
8	7.2	HTTPS on port 443 (default)	Remove any instances of http port in Cold Fusion templates and replace http port with the relative server.
9	7.7	WASS – Guest Checkbox	WASS will remove the guest checkbox on the Login interface. Have the system automatically recognized a guest user.
10	7.9	Remove Identity Type from the Participant Assessment Table.	Remove identity attribute from the column definition in the assessment table and replace the attribute with a stored procedure to find the sequential primary key value.
11	7.10	LOCCS/HUDCAPS Storing Data.	Remove storing Line of Credit Control System/ HUD Central Accounting Processing System (LOCCS/HUDCAPS) data in permanent tables and pipe the HOCCS/HUDCAPS data directly into the REAC database.
12	8	Modify the Review Submission Page description text.	On the Review Submission Page under the Financial Statement tab the instructions need to be modified. The first bullet of the instructions will be separated into two distinct bullets. The first bullet will read, "Government-Wide Financial Statements, If Applicable." The second bullet will read, "Fund Financial Statements."

1.3 System Overview

The FASS-PH is a subsystem of the Real Estate Assessment Center System (REACS). FASS-PH will help enable centralized financial analysis that can be used to identify where HUD should focus its limited resources to improve service delivery and manage its housing programs proactively. To achieve this goal, the following objectives have been identified:

- Gather standard financial data pertaining to each Public Housing Agency (PHA) and Section 8 Entity by combining standard fiscal audit information with reporting and compliance factors as defined by the Single Audit Act;
- Assess the financial condition of all PHAs and Section 8 Entities using a comprehensive protocol;
- Assess financial risk using standard financial data;
- Determine an objective, numerical score for each PHA and Section 8 Entity using standard protocols for financial performance review;
- Enable HUD staff to focus on the most troubled PHAs and Section 8 Entities based on the risk associated with the score;
- Eliminate or address existing material weaknesses identified through IG Audits. This includes mitigating potential risks;
- Support HUD's mission;
- Implement OMB Circular A-123 compliant policies and procedures;
- Support HUD's eGov Strategic Plan;
- Automate paper based forms to support the Government Paperwork Elimination Act (GPEA);
- Provide payback as early in the system lifecycle as possible;
- Provide significant benefits to HUD;
- All new functionality meets the Rehabilitation Act Section 508 requirements.

System Overview	
System and Subsystem Description	
System	Real Estate Assessment Center System (REACS)
Subsystem	Financial Assessment Subsystem - Public Housing (FASS-PH)
Responsible Party Description	
Sponsor	Public and Indian Housing – Real Estate Assessment Center (PIH-REAC)
Requirements	Avineon Inc.
Design	Avineon Inc.
Development	Avineon Inc.
System and Integration Testing	Avineon Inc., DCG
User Acceptance Testing	To be determined by PIH-REAC Management
Deployment	Avineon Inc., DCG
Maintenance	Avineon Inc., DCG
System Environment, Code, and Category: and Operational Status Description	
PCAS	307820
System Code	P093
System Category	Non-Major
Operational Status	Operational
System Environment	Web Based

The following table identifies and briefly describes the different users of FASS-PH.

User Environment	
Public Housing Agency (PHA)	PHAs and Section 8 Entities use FASS-PH to annually submit unaudited and audited financial data to the REAC for analysis. A PHA may also use the system to request submission extensions and to review the status of its financial submission(s).
Agents	A PHA may delegate the task of submitting financial information to its agent(s). Regardless of who submits its financial information, the PHA is ultimately responsible for the timeliness and accuracy of the submission.
Independent Public Accountant (IPA)	An IPA is responsible for reviewing and verifying audited submission data prior to a PHA's submission to the REAC. Again, the PHA is ultimately responsible for the timeliness and accuracy of the submission.
REAC PHA Financial Assessment Team	The REAC PHA Financial Assessment Team will use FASS-PH to perform its critical business processes, including the assessment of the financial condition of PHAs.

User Environment	
Other HUD Users	All other HUD users will have read-only access to FASS-PH in order to review PHA financial data. This includes field office and other HUD headquarters users.

1.4 Project References

The following documents provide a comprehensive understanding of the PHA financial assessment process. Most documents are available via the REAC Document Library. Additionally, several of the documents listed below are available through the PHA Financial Assessment Internet site at ***http://www.hud.gov/offices/reac/products/prodpha.cfm***

References
FASS-PH 8.1.0.0
"FASS-PH Release 8.1.0.0 SDM Design Phase – System Subsystem Specifications Document," 06/23/2005
"FASS-PH Release 8.1.0.0 SDM Design Phase – Database Specifications Document," 06/09/2005
"FASS-PH Release 8.1.0.0 SDM Define Phase – Functional Requirements Document," 06/07/2005
"FASS-PH Release 8.1.0.0 SDM Initiate Phase – Feasibility Study," 06/10/2005
"FASS-PH Release 8.1.0.0 SDM Initiate Phase – Cost/Benefit Analysis," 06/10/2005
"FASS-PH Release 8.1.0.0 SDM Initiate Phase – Risk Analysis," 06/09/2005.
"FASS-PH Release 8.1.0.0 SDM Initiate Phase – Risk Management Plan," 05/31/2005
"FASS-PH Release 8.1.0.0 SDM Initiate Phase – Project Management Plan," 05/31/2005
"FASS-PH Release 8.1.0.0 SDM Initiate Phase – Quality Assurance Plan," 05/31/2005
FASS-PH 8.0.0.0
"FASS-PH Release 8.0.0.0 SDM Functional Requirements Document," 11/13/2003.
"FASS-PH Release 8.0.0.0 SDM Data Requirements Document," 11/13/2003.
FASS-PH 7.4.0.0
"FASS-PH Release 7.4.0.0 SDM Integration Test Results and Evaluation Report," 10/31/2003.
"FASS-PH Release 7.4.0.0 SDM System Test Results and Evaluation Report," 10/10/2003
"FASS-PH Release 7.4.0.0 SDM Validation, Verification and Testing Plan," 09/04/2003.
"FASS-PH Release 7.4 0.0 SDM Program Specifications," 08/21/2003.
"FASS-PH Release 7.4.0.0 SDM System/ Subsystem Specifications,"08/21/2003.
"FASS-PH Release 7.4.0.0 SDM Functional Requirements Document," 08/05/2003.
"FASS-PH Release 7.4.0.0 SDM Data Requirements Document," 08/05/2003.
FASS PH 7.3.0.0
"FASS-PH Release 7.3.0.0 SDM Integration Test Results and Evaluation Report," 08/05/2003.
"FASS-PH Release 7.3.0.0 SDM System Test Results and Evaluation Report," 07/11/2003.
"FASS-PH Release 7.3.0.0 SDM Unit Test Plan," 05/27/2003.
"FASS-PH Release 7.3.0.0 SDM Validation, Verification and Testing Plan," 5/12/2003.
"FASS-PH Release 7.3.0.0 SDM Training Plan," 04/17/2003.
"FASS-PH Release 7.3 0.0 SDM Program Specifications," 05/07/2003.

References
"FASS-PH Release 7.3.0.0 SDM Database Specifications," 05/07/2003.
"FASS-PH Release 7.3.0.0 SDM System/ Subsystem Specifications," 05/07/2003.
"FASS-PH Release 7.3.0.0 SDM Functional Requirements Document," 04/02/2003.
"FASS-PH Release 7.3.0.0 SDM Data Requirements Document," 04/02/2003.
"FASS-PH Release 7.3.0.0 SDM System Security and Privacy Plan," 04/02/2003.
"FASS-PH Release 7.3.0.0 SDM System Support and Acquisition Document," 04/02/2003.
"FASS-PH Release 7.3.0.0 SDM Initiate Phase – Feasibility Study," 03/25/2003.
"FASS-PH Release 7.3.0.0 SDM Initiate Phase – Cost/Benefit Analysis," 03/25/2003.
"FASS-PH Release 7.3.0.0 SDM Initiate Phase – Needs Statement," 03/25/2003.
"FASS-PH Release 7.3.0.0 SDM Initiate Phase – Risk Analysis," 03/25/2003.
"FASS-PH Release 7.3.0.0 SDM Initiate Phase – Project Plan," 03/25/2003.
"FASS-PH Release 7.3.0.0 Work Plan," 03/20/2003.
"FASS-PH Quality Control Plan," 11/04/2002.
Refer to "FASS-PH Release 7.3.0.0 SDM Functional Requirements Document," 04/02/2003 for list of References prior to FASS-PH Release 7.3.0.0.
Policies
"PHAS: Physical Condition Scoring Process and Financial Condition Scoring Process," 10/21/2003.
"Changes to the Public Housing Assessment System (PHAS); Proposed Rule," 24 CFR Part 902, 02/06/2003.
"PHAS; Notice Adopting Interim Scoring Methodologies for PHAS Physical Condition and Financial Conditions Indicators," 03/15/2002.
"PHAS Information About PHAS Interim Scoring Methodology for PHAs With Fiscal Years Ending On or After September 30, 2001: Introduction; Notice," 11/26/2001.
"PHAS; Financial Condition Scoring Process Interim Assessments," 11/26/2001.
"PHAS; Revised Timetable for Issuance of Management Operations Official Scores and PHAS Advisory Scores; and Notice of Intent to Commence Informal Meetings on PHAS," 05/30/2001.
"PHAS; Financial Condition Scoring Process Notice," 12/21/2000.
"PHAS: Notice of Extended Submission Period for PHAS Management Operations Certification and Audited Financial Statement for Certain PHAs;" 11/21/2000.
"PHAS Management Operations Certification Resubmissions Period and Financial statement Submission Extension Period for Certain PHAs," 08/09/2000.
"PHAS Financial Condition Scoring Process," 06/28/2000.
"Uniform Financial Reporting Standards: 24 CFR Part 5, et al," 03/27/2000.
"Technical Correction to PHAS Final Rule," 06/06/2000.
"Public Housing Assessment System (PHAS) Amendments; Final Rule," 24 CFR Part 902, 01/11/2000.
"Notice Clarifying Manual Submissions and Extension Requests Under the Public Housing Assessment System (PHAS)," 11/17/1999.
"PHAS Proposed Amendments to 24 CFR Part 902," 06/22/1999.
"Public Housing Assessment System; Financial Condition Scoring Process Notice," 06/23/1999.
"Uniform Financial Reporting Standards for HUD Housing Programs; Final Rule," 24 CFR Part 5, et. al., 09/1/1998.

References
"Public Housing Assessment System Final Rule," 24 CFR Parts 901 and 902, 09/1/1998.
"HUD Handbook 2400.15," 02/18/1992.
"HUD Handbook 2229.1," 06/28/1989.
"HUD Handbook 2400.24 – Rev 2," 11/10/1999.
Additional References
OMB: "Information Collection; Request for Public Comments." 08/15/2003
"Federal Audit Clearinghouse (FAC) Summary of Proposed Changes to the Data Collection Form (SF-SAC)," 08/15/2003.
"Draft Data Collection Form (SF-SAC) for Fiscal Year Ending Dates in 2004, 2005, or 2006," 08/15/2003.
"Instructions for Completing Form SF-SAC, … for Fiscal Periods Ending in 2004, 2005, or 2006," 08/15/2003.
"Summary of Changes to SF SAC," 11/16/2000.
"I-TIPS FY2001 – Project Plan," 06/16/2000.
"I-TIPS FY2001 – Feasibility Study," 06/16/2000.
"I-TIPS FY2001 – Cost/Benefit Analysis," 06/16/2000.
"I-TIPS FY2001 – Needs Statement," 06/16/2000.
"I-TIPS FY2001 – Risk Assessment," 06/16/2000.
"Financial Data Schedule Line Definitions and Crosswalk Guide," 09/14/2001.
"HUD PHA GAAP Conversion Guide," 01/31/2000.
"Detailed System Requirements Document for the AFS Version 2.0."
"Annual Financial Data Submission Requirements for the AFS Version 2.0."
"Addendum to the Data Standardization Results for the AFS Version 2.0."
"System Development Methodology Release 6.01," January 2000.
"Preliminary Scoring Methodology and Thresholds for Financial Indicators," 06/30/1999.
"Financial Indicators Methodology & Analysis Guide," 12/14/1999.
"PHA Financial Assessment Lab Financial Assessment Operations Design and Procedures," 03/31/1999.
"Financial Assessment Lab – Business Process Documentation and Flow Maps," 09/21/1999.
"PHAS Appeals Business Process," 11/28/2000.
"HUD Business Resumption Plan," 10/2000.

1.5 Terms and Abbreviations

The following table defines terms and acronyms used throughout this document.

Term	Definition
ACWP	Actual Cost of Work Performed
APP	Annual Performance Plan
BCWP	Budgeted Cost of Work Performed
BCWS	Budgeted Cost of Work Scheduled
BLOBs	Binary Large Objects

Term	Definition
BRD	Business Requirements Document
CCB	Change Control Board
CCD	Change Control Board
CDR	Critical Design Review
CFDA	Catalog of Federal Domestic Assistance
CI	Configuration Item
CIO	Chief Information Officer
CLIN	Contract Line Item Number
CM	Configuration Management
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integrated
CMP	Configuration Management Plan
CO	Contracting Office
COR	Contracting Office Representative
COTS	Commercial Off The Shelf
CPI	Cost Performance Index
CR	Change Request
CSCI	Computer Software Configuration Item
CV	Cost Variance
DB	Database
DCF	Data Collection Form
DCG	Development Coordination Group
DMM	Deliverable Management Module
DOA	Date of Award
DR	Design Review
EAC	Estimate At Completion
EIN	Employer Identification Number
ETC	Estimate To Complete
EV	Earned Value
EVA	Earned Value Analysis
EVM	Earned Value Management
FASS	Financial Assessment Subsystem
FASS-PH	Financial Assessment Subsystem – Public Housing
FCA	Functional Configuration Audit
FDS	Financial Data Schedule
FEDSIM	Federal Systems Integration and Management Center
FOIA	Freedom Of Information Act
FQR	Formal Qualification Review
FRD	Functional Requirements Document
FY	Fiscal Year
FYE	Fiscal Year End
GAAP	Generally Accepted Accounting Principles

Term	Definition
GAGAS	Generally Accepted Government Auditing Standards
GAO	Government Accounting Office
GASB	Governmental Accounting Standards Board
GPEA	Government Paperwork Elimination Act
GSA	General Services Administration
GTM	Government Technical Monitor
HA(s)	Housing Authority
HTML	Hypertext Markup Language
HUD	Department of Housing and Urban Development
HUD OIG	HUD Office of Inspector General
HUDCAPS	HUD Central Accounting Processing System
HUDWeb	HUD's Intranet Web Site
ICD	Interface Control Deliverable
IG	Inspector General
IG	Inspector General
IPA	Independent Public Accountant
IPR	In Progress Reviews
ISG	Internet Services Group
IT	Information Technology
IV&V	Independent Verification & Validation
JAD	Joint Application Development
LOCCS	Line of Credit Control System
LPF	Late Presumptive Failure
MF	Multi-Family
NASS	Integrated Assessment Subsystem
NDS	Non-Developmental Software
ODC	Other Direct Costs
OMB	Office of Management and Budget
PASS	Physical Assessment Subsystem
PCA	Physical Configuration Audit
PD&R	Policy Development and Research
PDR	Preliminary Design Review
PH	Public Housing
PHA	Public Housing Agency/Public Housing Authority
PHAS	Public Housing Assessment System
PIH	Public and Indian Housing
PIH-REAC	Public Indian Housing - Real Estate Assessment Center
PM	Project Manager
PMC	Project Monitoring and Control
PMP	Project Management Plan
PNR	Problem Notification Report
POC	Points of Contact

Term	Definition
PP	Project Plan
PP&O	Project Planning & Oversight
PPQA	Product & Process Quality Assurance
PR	Problem Reports
PRR	Product Readiness Review
QA	Quality Assurance
QAG	Quality Assurance Guidelines
QAP	Quality Assurance Plan
QASS	Quality Assessment Subsystem
RAF	Risk Analysis Form
RASS	Residential Assessment Subsystem
REAC	Real Estate Assessment Center
REACS	Real Estate Assessment Center System
RM	Risk Management
RR	Requirements Review
SAC	PHAS invalidation action code
SCI	Software Configuration Item
SCR	Software Change Request
SDD	Software Design Description
SDF	Software Development File
SDL	Software Development Library
SDM	Software Development Methodology
SDP	System Decision Paper
SDR	Software Design Review
SMP	Software Measurement Plan
SOW	Statement Of Work
SPI	Schedule Performance Index
SQA	Software Quality Assurance
SQL	Standard Query Language
SR	Specification Review
SRS	Software Requirements Specification
SSDD	System/Subsystem Specification
SSR	Software Specification Review
SSS	System/Subsystem Specification
SV	Schedule Variance
SW	Software
TAC	Technical Assistance Center (formerly the Customer Service Center)
TBD	To Be Defined
TOR	Task Order Request
TOS	Tracking & Ordering System
TRB	Technical Review Board

Term	Definition
TRR	Test Readiness Review
UAT	User Acceptance Testing
UDF	Unit Development Folder
UFI	Unique Fee Accountant Identifier
UFRS	Unified Financial Recording Standards
UII	Unique IPA Identifier
WASS	Web Access Security System
WBS	Work Breakdown Structure
WDDX	Web Dynamic Exchange
XML	eXtensible Mark-up Language

1.6 Points of Contact

1.6.1 Information

Points of Organizational Contacts Table				
Contact Name	**Organization**	**Position**	**Telephone Number**	**Email**
Nick Miele	PIH-REAC	FASS PH Business Program Manager	202-475-8788	Nicholas_X._Miele@hud.gov
Steve Bolden	PIH-REAC	FASS PH Assessment Manager for Systems Operations	202-475-8706	Steve_A._Bolden@hud.gov
Freddie Harrison	PIH-REAC	FASS PH IT Manager	202-475-8639	Frieddie_Harrison@hud.gov
Keith Bennett	Avineon Inc.	Project Manager	202-475-8903	Keith_Bennett@HUD.gov
Joneff Chung	Avineon Inc.	FASS-PH Requirements Lead	202-475-8889	Joneff_Chung@HUD.gov
Surafiel Berek	Avineon Inc.	FASS-PH Development Lead	202-475-8828	Surafiel_Berek@HUD.gov
Mohammed Hasan (Ashraf)	Avineon Inc.	FASS-PH Maintenance Lead	202-475-8898	Mohammed_A._Hasan@HUD.gov

1.6.2 Coordination

FASS-PH will coordinate with the following organizations to successfully implement the FASS-PH Release 8.1.0.0:

Organization	Support Function

Organization	Support Function
PIH-REAC	Business Requirements Support, Project Management
Avineon/FTI/Mil	Requirements, Design, Development, Testing, Installation, Deployment, Maintenance, Technical Support /Operations, Project Management
DCG	Customer Support/Operations, Development Coordination, Integration Test Coordination, Deployment, and Maintenance
HUD IT	Implementation Coordination
FASS-PH Lab	Business Requirements Support, Version 4 Scoring Thresholds
WASS	Security
NASS	Integration
QASS	Integration

Listed below are the coordination dates that each Real Estate Assessment Center System (REACS) must achieve in order to adhere to the August 26, 2005 release date.

Integration Test Start Date	Code Locked Date	Integration Test End Date	Final HARTS Request Submission Date	Release Date
07/13/2005	08/01/2005	08/08/2005	08/01/2005	08/26/2005

1.7 Master List of Programs

The following is a list of programs that will be implemented in FASS PH Release 8.1.0.0. These programs will either be new for the system or modifications of existing functionality. This information will be clearly stated within the content of the program descriptions.

1.7.1 DCF Financial Statement Line Item G3000-010

The DCF/Financial Statement/G3000-010 Type of Audit Report/G3000-060 & G3000-070 text will be modified to reflect opinion of Fund (rather than Program). Auditors should only be entering an opinion for funds with in the PHA.

1.7.2 View of Prior Years Submission Comments

The Review Submission "View Prior Fiscal Years Comments Page" provides FASS-PH internal users the ability to review prior years reviewers comments while still reviewing the current year's submissions.

This will allow the reviewers to review prior submission comments without navigating between multiple submissions.

1.7.3 Additional Valid Value for DCF Line Item 4200-050

A new logic will be introduced to Line Items: G4200-050 and G4200-010. If the external user selects a "No" value for Line Item G4200-010, Line Item G4200-050 will default to "Not Applicable" value and a pop-up window will inform the user of the logic between these two Line Items.

1.7.4 Line Item 1102

New logic will be in place, so that the external user will not be able to enter any amount on Line Item 1102. This new methodology should begin for all 9/30/2005 submissions.

1.7.5 Analyst Column Display of Analyst and Manager's Names

External Inbox of the FASS-PH analyst column modification is a maintenance requirement and will introduce a new logic regarding how to display the FASS-PH employee in the FASS-PH Analyst column. Only the financial analyst or manager that last view the submission will be display in this column. The director's name must not appear in this column.

1.7.6 FDS Report Page Print Functionality

This requirement will place page breaks on the Financial Data Schedule (FDS) report, so that it prints properly.

1.7.7 Review Submission Page Text Change

The descriptive text on the Review Submission Page under the Financial Statement tab will be modified to read "Government-Wide and or Fund Financial Statement."

1.8 Infrastructure Changes

The following infrastructure enhancements will be implemented along with Release 8.1.0.0 as mandated by the Development Coordination Group (DCG).

1.8.1 File Attachments Storage

This is a modification to how file attachments are stored in FASS-PH. All file attachments in FASS-PH assessments must now be stored as Binary Large Objects (BLOBs) in the database.

This functionality will replace the current method of storing files in directories within the FASS-PH. Prior to release 8.1.0.0 DCG will move all attachments to the REAC database.

1.8.2 Application Use of Port 443

All hard coded port number from all Cold Fusion templates and replace the port number with the relative server name.

1.8.3 Removal of Check Box for Guest Users

The Removal of Check Box requirement calls for modification to how a guest user logs into FASS-PH. Currently a guest user manually checks the Guest User checkbox on the login interface. Web Access Subsystem (WASS) will remove the Guest user checkbox beginning with Release 8.1.0.0. The FASS-PH system will implement a new functionality on how a guest user is identified. The system will search to find if any Role ID's is associated with the User ID. If a Role ID's is associated with the User ID, the user will be redirected to the Main Internal page. If no Role ID is associated with the User ID, the user will be redirected to the Main Guest Page.

1.8.4 Identity Type

The Identity Type requirement calls for modification to the participant assessment table in the REAC database. The identity attribute of the column definition in a database is used by the database application to find the next sequential primary key. The identity attribute of the column definition will be disabled for the assessment_id field in the participate_assessment table. To compensate for the lack of the identity functionality, the maximum assessment_id is identified and will be incremented by one.

1.8.5 LOCCS/HUDCAPS Interface Change

A new procedure will be implemented on how data is inserted into the LOCCS/HUDCAPS modification to the LOCCS/HUDCAPS interface. According to DCG requirements, applications must not create permanent tables within HUD related databases. A new procedure will be created to insert LOCCS/HUDCAPS data into the REAC database.

2.0 DCF Financial Statement Line Item G3000-010

2.0 DCF FINANCIAL STATEMENT LINE ITEM G3000-010

2.1 Program Description

The DCF/Financial Statement/G3000-010 Type of Audit Report/G3000-060 and G3000-070 text will be modified to reflect opinion of Fund (rather than Program). Auditors should only be entering an opinion for funds with in the PHA.

New or Modification: Modified

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
1.1	1.4.1	494	RTM

2.1.1 Software Unit Description

Text changes are needed to adhere to accounting principles. Accountants should only be entering opinion of funds within the PHA and not entering opinion of a program. The text changes will occur in the external and internal DCF Federal Program interface and in the Line Item G3000-010 Detail interface.

The text will change in only one place on the external and internal DCF Federal Program interface. In the description column on the Line Item G3000-010 row, the text will change from "Type of Audit Report" to "Fund Opinion(s)".

For Line Item G3000-060 and Line Item G3000-070 Detail interface, the text will change for every Federal Program name. For each Federal Program name that occurs on this interface, the wording "Fund Type and Opinion of the Fund containing" will be placed in front of the Federal Program name.

2.1.2 Software Unit

The page is accessible by selecting the Financial Statement tab on the FASS-PH system. Users access the DCF Federal Program interface and the Line Item G3000-010 and click on the "details" interface link to view Line Items G3000-060 and G3000-070. The text change will be visible for each Federal Program name that occurs on the interface.

2.1.3 Accuracy and Validity

DCF/Financial Statement line item G3000-010 Type of Audit Report/G3000-060 and G3000-070 "Opinion of the Fund" will be reflected rather the Program.

2.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

2.1.5 Adaptability

Flexibility of FASS-PH will be enhanced by this requirement and text changes are needed to adhere to accounting principles.

2.2 Environment

2.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

2.2.2 Interfaces

The DCF Financial Statement Line Item G3000-010 does not interface with other systems.

2.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

2.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

2.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

2.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

2.2.3.4 Temporary and Permanent Storage

There are no new temporary and permanent storage requirements to support this capability.

2.2.4 Security

The following table shows user access rights for the DCF/Financial Statement reports.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

2.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

2.3 Design Details

The following screenshots are prototypes for DCF Financial Statement line item G3000-010

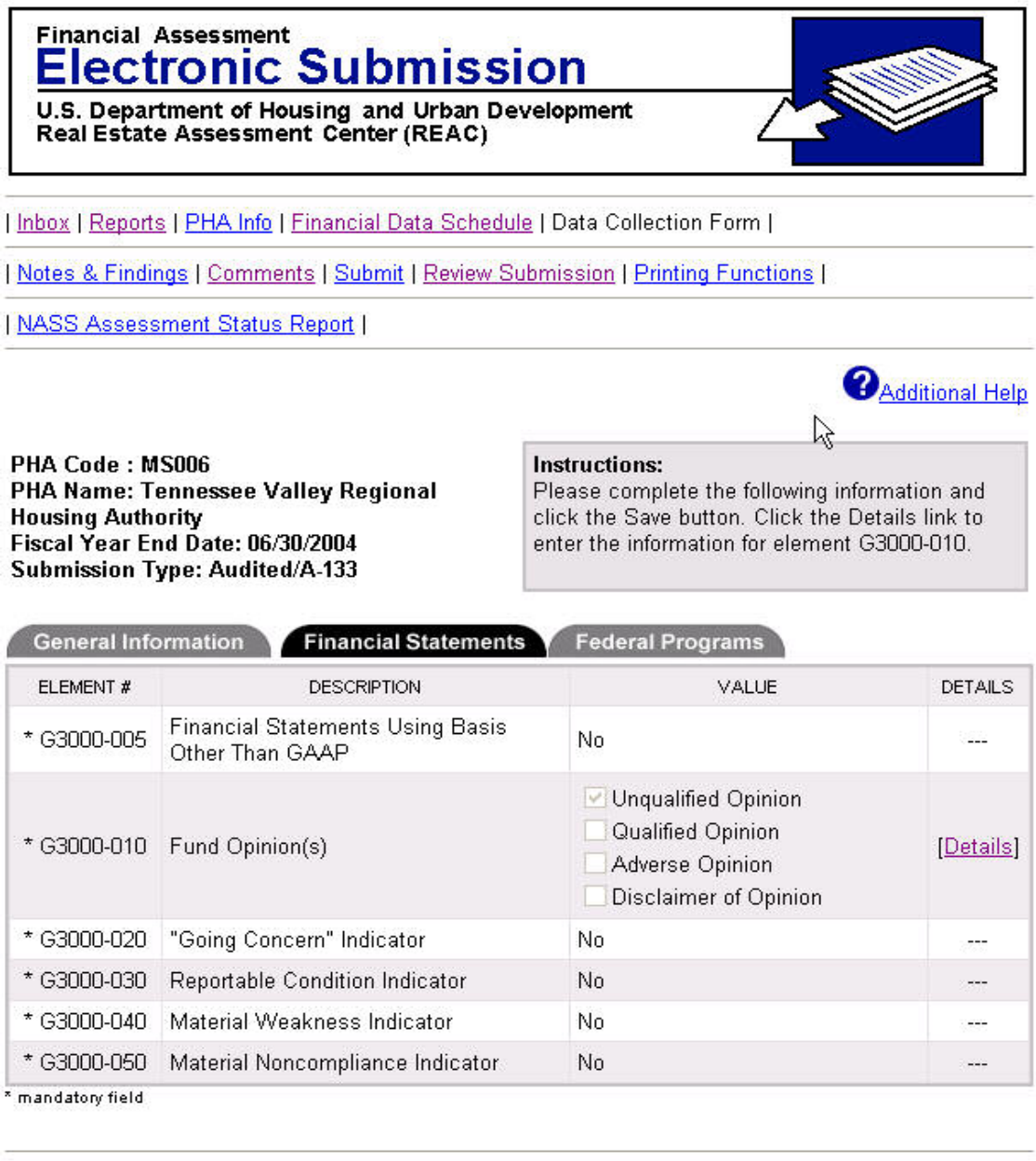


Figure 2-1: FINANCIAL STATEMENT (ELEMENT # G3000-10)

| Inbox | PHA Info | Financial Data Schedule | Data Collection Form | Notes & Findings |

| Comments | Submit | Review Submission | Printing Functions | NASS Assessment Status Report |

Additional Help

PHA Code : MS006
PHA Name: Tennessee Valley Regional Housing Authority
Fiscal Year End Date: 06/30/2004
Submission Type: Audited/A-133

Instructions:
Please select the Fund Type and Fund Opinion for each program and click the Save button. All non-major funds must have the same Fund Opinion.

Audit Details [Back to Financial Statements]

CFDA#	NAME OF PROGRAM		DETAILS
14.182	**Fund Type and Opinion of the Fund containing N/C S/R Section 8 Programs**		---
* G3000-060	Fund Type	Major Fund	---
* G3000-070	Fund Opinion	Unqualified Opinion	[Details]
14.850a	**Fund Type and Opinion of the Fund containing Low Rent Public Housing**		---
* G3000-060	Fund Type	Major Fund	---
* G3000-070	Fund Opinion	Unqualified Opinion	[Details]
14.870	**Fund Type and Opinion of the Fund containing Resident Opportunity and Supportive Services**		---
* G3000-060	Fund Type	Non Major Fund	---
* G3000-070	Fund Opinion	Unqualified Opinion	[Details]
14.871	**Fund Type and Opinion of the Fund containing Housing Choice Vouchers**		---
* G3000-060	Fund Type	Major Fund	---

Figure 2 2: FINANCIAL STATEMENT (ELEMENT # G3000-60 & ELEMENT # G3000-70)

2.3.1 Input

No input records for this requirement

2.3.1.1 Input Records

The line items on the database are used as input to the process. There are no actual input records.

2.3.1.2 Input Data Elements

The line items on the database are used as input to the process.

2.3.2 Output

The text changes are shown on the screen.

2.3.2.1 Output Reports

The text changes are shown on the screen.

2.3.2.2 Output Data Element

2.3.3 Software Relationships

No changes for Release 8.1.0.0

2.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
G3000-010	No (REAC)	page_ref	Item_text	Varchar(255)	Fund Opinion(s)	It displays the description element number G3000-010
G3000-060	No (REAC)	program_ref	program_ref_name	Varchar(70)	Fund Type and Opinion of the Fund containing + program name	It displays the description element number G3000-060
G3000-070	No (REAC)	program_ref	program_ref_name	Varchar(70)	Fund Type and Opinion of the Fund containing + program name	It displays the description element number G3000-070

Software Unit Logic

Scripts:

Update the reference table to display Fund Opinion(s):

```
UPDATE page_ref
   SET item_text = 'Fund Opinion(s)'
If  item_id = 400211
```

To update the reference table value:

```
USE reacs
GO

DECLARE
@rc          INTEGER,
@reason       VARCHAR (255)

SELECT @rc = 0
SELECT @reason = "Successful item_text update for item_id 40021 for pag-ref table"


BEGIN TRANSACTION

UPDATE page_ref
SET item_text = 'Fund Opinion(s)'
WHERE item_id = 400211

IF @@error<>0
BEGIN
      SELECT @rc = 1, @reason = 'Error in updating item_text for item_id 40021 for page_ref table'
      GOTO SQL_ERROR_1
END


SQL_ERROR_1:

IF @rc > 0
      BEGIN
            SELECT "ROLLBACK TRANSACTION", @rc AS RC, @reason AS REASON
            ROLLBACK TRANSACTION
      END
ELSE
      BEGIN
            SELECT @rc AS RC, @reason AS REASON
```

```
            COMMIT TRANSACTION
                                SELECT 'RC=0'
      END
GO
```

3.0 PRIOR YEARS SUBMISSION COMMENTS REVIEW

3.0 PRIOR YEARS SUBMISSION COMMENTS REVIEW

3.1 Program Description

The prior years submission comments review capability will allow users to view prior PHA's Fiscal Year Submission Comments. A hyperlink will be incorporated to the "Reviewer Comments" interface to direct users to the "View Prior Years Comment" interface. This new interface will be divided into two sections. The first top section will be the interactive part of the interface. Users will have the ability to view prior years comments by selecting from two criteria, Fiscal Year and Submission type. Also, a "Review Comments" link will be part of this section to redirect users back to the "Reviewer Comments" interface. The second bottom section of the interface will display the output. The output will display a listing of all comments, their entry date, and their recommendation. By default, the interface will display all comments and their attributes for the selected fiscal year. If a PHA does not have comments from the previous FY year, the interface will indicate that fact.

New or Modification: New

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
6.1	1.4.2	504	RTM

3.1.1 Software Unit Description

The "View Prior Fiscal Years Comment" interface is a new functionality to the FASS-PH system. This interface will allow only the internal user, which consists of the Financial Analyst, Assessment Manager, and the Director to view prior fiscal years' comments.

This interface will be linked from the "Reviewer Comments" interface. Once the internal user selects the link, the browsers will redirect the internal user to the "View Prior Fiscal Years Comment" interface. This interface will be divided into two sections: an interactive section and an output section

The interactive section of this interface will provide the user with the ability to view different outputs by changing the search criteria. The search criteria are Fiscal Year and Submission Type. The Fiscal Years search criterion will have the date range of all submissions' fiscal years currently in the FASS-PH. The Submission Type search criterion consists of only two options of Audited and Unaudited submissions. By default, the search criteria values are the current fiscal year of the selected submission and both Audited and Unaudited submissions. In addition, the

user will have the ability to redirect the browser to the original 'Reviewer comment' interface by selecting the 'Back to Reviewer comment page link.

The output section of this interface will allow the user to view all of the comments and their attributes for the initial search criteria default values or the user's selection of the search criteria. The output will consists of the reviewer's name, comment(s), the comment's entry date(s), and the recommendation(s).

3.1.2 Software Unit

The page can be accessed from the "Reviewer Comments" page by clicking the "View Prior Fiscal Years Comment" link. The "Go Button" will allow a user to submit a fiscal year and submission type search criteria. This button will be displayed at all times on the 'View Prior Fiscal Years Comments' page.

3.1.3 Accuracy and Validity

By default, the interface displays all comments and their attributes from the selected year. If PHA does not have any comments from the previous year, the interface will indicate that fact.

3.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

3.1.5 Adaptability

FASS-PH will display a report of any edit flags generated by the submission to both external and internal users. This report will allow PHAs to modify their submissions in response to possible flags that would have been identified during the internal review process. In turn, this report will reduce the manual review processes for internal users. "Back to Reviewer Comment" page hyperlink allows the user to navigate to 'Reviewer Comments' page.

3.2 Environment

3.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.

Software	Description
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

3.2.2 Interfaces

Reviewing of prior years submission comments does not interface with other systems.

3.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

3.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

3.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

3.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

3.2.3.4 Temporary and Permanent Storage

There are no new temporary and permanent storage requirements to support this capability.

3.2.4 Security

The following table shows user access rights for reviewing prior years submission comments.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.

Role	Create	Read	Update	Delete	Detailed Description
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

3.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

3.3 Design Details

The following screenshots are prototypes for the submission comments review screen.

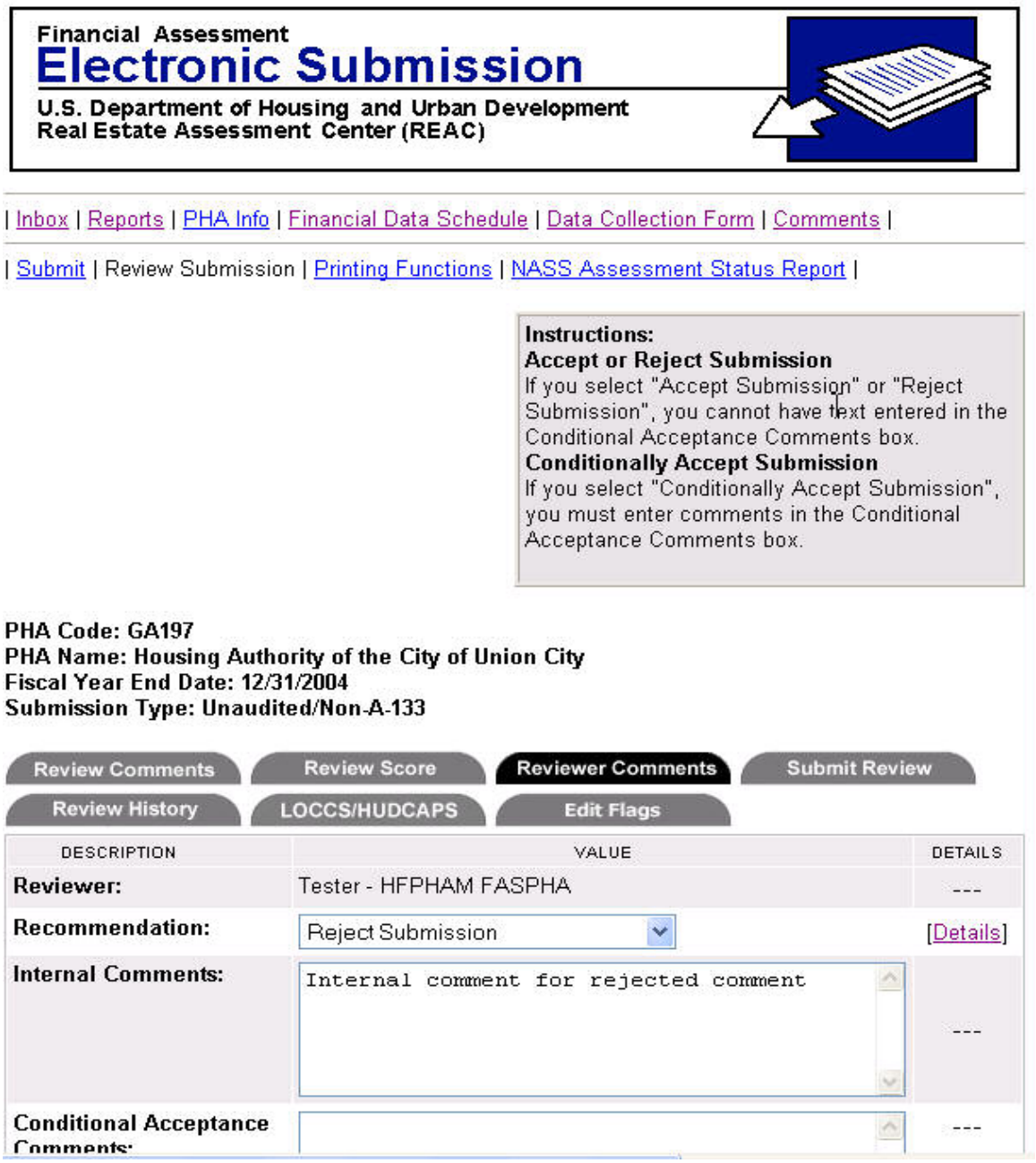


Figure 3-1: Internal Users – Interface for Prior Years Comments Review

Figure 3 2: Internal Users – Interface for Prior Years Comments Review

3.3.1 Input

Fiscal Year and Submission Type.

3.3.1.1 Input Records

There are no actual input records.

3.3.1.2 Input Data Elements

participant_id
fiscal_year
user_id

3.3.2 Output

The output section of this interface will allow the user to view all of the comments and their attributes for the initial search criteria default values or the user's selection of the search criteria. The output will consist of the reviewer's name, comment(s), the comment's entry date(s), and the recommendation(s).

3.3.2.1 Output Reports

Current year and prior years review comments, reviewer's name, and review date.

3.3.2.2 Output Data Elements

fiscal_year user_id
action_date
action-code
action_comment,

3.3.3 Software Relationships

The "View Prior Fiscal Years Comment" interface is a new functionality to the FASS-PH system. This interface will allow only the internal user, which consists of the Financial Analyst, Assessment Manager, and the Director to view prior fiscal years' comments.

This interface will be linked from the Reviewer Comments interface. Once the internal user clicks the link, the browsers will redirect the internal user to the "View Prior Fiscal Years Comment" interface. This interface will be divided into two sections: an interactive section and an output section

The interactive section of this interface will provide the user with the ability to view different outputs by changing the search criteria. The search criteria are "Fiscal Year" and "Submission Type".

The "Fiscal Years" search criterion will have the date range of all submissions' fiscal years currently in the FASS-PH.

The "Submission Type" search criterion will consists of only two options of Audited and Unaudited submissions. By default, the search criteria values will be the current fiscal year of the selected submission and both Audited and Unaudited submissions. Also the user will have the ability to redirect the browser to the original 'Reviewer comment' interface by clicking the 'Back to Reviewer' comment page link.

The output section of this interface will allow the user to view all of the comments and their attributes for the initial search criteria default values or the user's selection of the search criteria. The output will consists of the reviewer name, comment(s), the comment's entry date(s), and the recommendation(s).

3.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore / Reference Table	Data Element	Data Element Type/ Length	Values	Description/ Comments
Reviewer	No (REAC) (HERMS)	participant_assess_action/ Msf_user	User_id/ user_first_name/user_last_name	Char(6)	user_first_name/ user_last_name	It will display the first name and last name of the person who submitted the comment
Recommendation Type	No (REAC)	participant_assess_action	action_code	Char(3)	AP1, AP2, APS, RJ1, RJ2, RJS, AC1, AC2 and ACV	It displays the recommendation type. It will display 'Accept Submission' if the action code is 'AP1', 'AP2', and 'APS'. It will display 'Rejected Submission' if the 'RJ1', 'RJ2', and 'RJS'. It will display 'Conditionally Accept Submission' if the action code is AC1, AC2 and ACV
Comment	No REAC	participant_assess_action	action_comment	text	Action_comment value from participant_assess_action	It displays the internal comment for Conditional Acceptance, Accept Submission and Rejection comment.

Field Name	Editable by REAC (Source System)	Datastore / Reference Table	Data Element	Data Element Type/ Length	Values	Description/ Comments
Comment Date	(No) REAC	participant_assess_action	action_date	Datetime	action_date value fromparticipant_assess_action table	This is the date where the comment is inserted or updated into the database.

Software Unit Logic

View Prior Years Submission Comments Stored Procedures

To populate comments for the current fiscal year.

To identify the fiscal years:

```
SELECT distinct datename(yy,year_end_dt) as fiscal_year
FROM year_end_ref noholdlock
Where datename(yy,year_end_dt)  > '1998'
ORDER BY fiscal_year desc
```

To identify the assessments for the selected fiscal year:

```
SELECT participant_id, assessment_id, group_id,
version_id,  fiscal_year
FROM participant_assessment noholdlock
WHERE participant_id = @participant_id
AND group_id IN (56, 57)
AND fiscal_year = @fiscal_year
```

Once the assessment_id(s) are identified, identify the internal comments.

```
SELECT action_code, action_comment,
convert(varchar,action_date,101) as action_date
FROM participant_assess_action noholdlock
WHERE object_id = 1 and
group_id = #group_id# and
assessment_id = # assessment_id#
and participant_id = # participant_id#
and system_id = 'FASPHA'
and action_code in ('AC1','AC2','ACV', 'AP1', 'AP2', 'APS', 'RJ1', 'RJ2', 'RJS)
```

1. faspha_prior_fy_comment:

```
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_prior_fy_comment')
  BEGIN
    PRINT 'DROP PROCEDURE faspha_prior_fy_comment'
    DROP PROCEDURE faspha_prior_fy_comment
  END
GO
CREATE PROCEDURE faspha_prior_fy_comment
(
 @participant_id  NUMERIC(10,0),
```

```
 @assessment_id  NUMERIC(9,0),
 @fiscal_year   NUMERIC(4,0)
)
AS

IF @@TRANCOUNT = 0
SET CHAINED OFF

DECLARE @rc   INTEGER,
  @reason  VARCHAR(255)

SELECT  @rc = 0,
  @reason = NULL

CREATE TABLE #priorcomment
(
  participant_id NUMERIC(10),
  assessment_id  NUMERIC(9,0),
  group_id  SMALLINT,
  version_id SMALLINT,
  fiscal_year NUMERIC(4,0),
)
IF (@@error <> 0)
  BEGIN
   SELECT @rc = 1,
        @reason = "Failed to create temp table"
    GOTO SQL_ERROR
END

INSERT INTO #priorcomment
(
  participant_id,
  assessment_id,
  group_id,
  version_id,
  fiscal_year
)
SELECT participant_id, assessment_id, group_id,
version_id,  fiscal_year
FROM participant_assessment noholdlock
WHERE participant_id = @participant_id
AND group_id IN (56, 57)
AND fiscal_year = @fiscal_year
IF (@@error <> 0)
   BEGIN
```

```
 SELECT @rc = 1,
 @reason="Failed while inserting data into temp table"
 GOTO SQL_ERROR
END

SELECT  paa.action_code, paa.action_comment, paa.user_id,
    convert(varchar,paa.action_date,101) as action_date
FROM    participant_assess_action paa NOHOLDLOCK,
    #priorcomment pr
 WHERE paa.participant_id = @participant_id
   AND paa.group_id = pr.group_id
   AND paa.assessment_id = pr.assessment_id
   AND paa.version_id = pr.version_id
   AND (paa.action_code in ('AC1','AC2','ACV')
   OR  paa.action_code in ('AP1', 'AP2', 'APS', 'RJ1', 'RJ2', 'RJS'))
   AND paa.system_id = 'FASPHA'
   AND paa.object_id = 1
   ORDER BY action_date desc
IF (@@error <> 0)
  BEGIN
 SELECT @rc = 1,
 @reason="Failed to select participant action"
 GOTO SQL_ERROR
END

SELECT distinct datename(yy,year_end_dt) as fiscal_year
FROM year_end_ref noholdlock
WHERE datename(yy,year_end_dt)  > '1998'
ORDER BY fiscal_year desc
IF (@@error <> 0)
BEGIN
  SELECT @rc = 1,
       @reason="Failed to select fiscal year"
   GOTO SQL_ERROR
 END

SQL_ERROR:
IF @@error <> 0
      BEGIN
            SELECT 'ROLLBACK TRANSACTION', @rc AS RC, @reason AS REASON
            ROLLBACK TRANSACTION
      END
ELSE
      BEGIN
            SELECT @rc AS RC, @reason AS REASON
```

```
            COMMIT TRANSACTION
       END
GO
GO
IF EXISTS (SELECT * FROM sysobjects WHERE name ='faspha_prior_fy_comment' AND
type='P')
  BEGIN
        PRINT 'STORED PROCEDURE : faspha_prior_fy_comment - successfully added'
     EXEC sp_help faspha_prior_fy_comment
  END
ELSE
  BEGIN
        PRINT 'STORED PROCEDURE : faspha_prior_fy_comment - NOT added'
  END
    GO
```

2. faspha_prior_fy_comment2:

```
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_prior_fy_comment2')
 BEGIN
   PRINT 'DROP PROCEDURE faspha_prior_fy_comment2'
   DROP PROCEDURE faspha_prior_fy_comment2
 END
GO
CREATE PROCEDURE faspha_prior_fy_comment2
(
 @participant_id  NUMERIC(10,0),
 @assessment_id  NUMERIC(9,0),
 @fiscal_year   NUMERIC(4,0),
 @group_id     SMALLINT
)
AS

IF @@TRANCOUNT = 0
SET CHAINED OFF

DECLARE @rc   INTEGER,
  @reason  VARCHAR(255)

SELECT  @rc = 0,
  @reason = NULL

CREATE TABLE #priorcomment2
```

```
(
  participant_id NUMERIC(10),
  assessment_id  NUMERIC(9,0),
  group_id  SMALLINT,
  version_id SMALLINT,
  fiscal_year NUMERIC(4,0)
)
IF (@@error <> 0)
  BEGIN
   SELECT @rc = 1,
        @reason = "Failed to create temp table"
    GOTO SQL_ERROR
END
-- grab assessment ids
INSERT INTO #priorcomment2
(
  participant_id,
  assessment_id,
  group_id,
  version_id,
  fiscal_year
)
SELECT participant_id, assessment_id, group_id,
version_id,  fiscal_year
FROM participant_assessment noholdlock
WHERE participant_id = @participant_id
AND group_id = @group_id
AND fiscal_year = @fiscal_year
ORDER BY assessment_id desc
IF (@@error <> 0)
   BEGIN
 SELECT @rc = 1,
 @reason="Failed while inserting data into temp table"
 GOTO SQL_ERROR
END

SELECT  paa.action_code, paa.action_comment, paa.user_id,
    convert(varchar,paa.action_date,101) as action_date
FROM    participant_assess_action paa NOHOLDLOCK,
    #priorcomment2 pr
 WHERE paa.participant_id = @participant_id
   AND paa.group_id = @group_id
   AND paa.assessment_id = pr.assessment_id
   AND paa.version_id = pr.version_id
   AND (paa.action_code in ('AC1','AC2','ACV')
```

```
 OR  paa.action_code in ('AP1', 'AP2', 'APS', 'RJ1', 'RJ2', 'RJS'))
   AND paa.system_id = 'FASPHA'
   AND paa.object_id = 1
   ORDER BY action_date desc
IF (@@error <> 0)
   BEGIN
 SELECT @rc = 1,
 @reason="Failed to select participant action"
 GOTO SQL_ERROR
END

SELECT distinct datename(yy,year_end_dt) as fiscal_year
FROM year_end_ref noholdlock
WHERE datename(yy,year_end_dt)  > '1998'
ORDER BY fiscal_year desc
IF (@@error <> 0)
BEGIN
   SELECT @rc = 1,
        @reason="Failed to select fiscal year"
    GOTO SQL_ERROR
 END

SQL_ERROR:
IF @@error <> 0
       BEGIN
             SELECT 'ROLLBACK TRANSACTION', @rc AS RC, @reason AS REASON
             ROLLBACK TRANSACTION
       END
ELSE
       BEGIN
             SELECT @rc AS RC, @reason AS REASON
             COMMIT TRANSACTION
       END
GO
GO
IF EXISTS (SELECT * FROM sysobjects WHERE name ='faspha_prior_fy_comment2' AND
type='P')
   BEGIN
        PRINT 'STORED PROCEDURE : faspha_prior_fy_comment2 - successfully added'
     EXEC sp_help faspha_prior_fy_comment2
   END
ELSE
   BEGIN
        PRINT 'STORED PROCEDURE : faspha_prior_fy_comment2 - NOT added'
   END
```

```
GO
```

3. faspha_select_username:

```
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_select_username')
  BEGIN
    PRINT 'DROP PROCEDURE faspha_select_username'
    DROP PROCEDURE faspha_select_username
  END
GO

CREATE PROCEDURE faspha_select_username
(
        @user_id        CHAR(6)
)


AS

IF @@TRANCOUNT = 0
SET CHAINED OFF

DECLARE     @rc     INTEGER,
            @reason       VARCHAR(255)

SELECT        @rc = 0,
            @reason = NULL

SELECT user_first_name + ' ' + user_last_name  as user_name
        FROM mfs_user a noholdlock
        WHERE user_id = @user_id

IF @@error <> 0 GOTO SQL_ERROR


GOTO SQL_ERROR

SQL_ERROR:

IF @@error > 0
        BEGIN
                SELECT @rc = 12, @reason = 'Failure in SQL Statement'
                SELECT @rc AS RC, @reason AS REASON
                RETURN- 12
```

```
        END
ELSE
        BEGIN
                SELECT @rc AS RC, @reason AS REASON
                RETURN 0
        END

GO
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_select_username')
  BEGIN
    PRINT 'faspha_select_username SUCCESSFUL'
    EXEC sp_help faspha_select_username
  END
ELSE
  BEGIN
    PRINT 'CREATE faspha_select_username FAILED'
  END
GO
```

4.0 ADDITIONAL VALID VALUE FOR DCF LINE ITEM 4200-050

4.0 ADDITIONAL VALID VALUE FOR DCF LINE ITEM 4200-050

4.1 Program Description

DCF Line Item 4200-050 will be defaulted to "N/A" for Non-Major Programs audited A133, there will be no penalty when this opinion is selected. If a user selects 'No' for line item 4200-010, the user will be alerted and line item 4200-050 will be defaulted to 'N/A'

New or Modification: Modification

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
1.1	1.4.1	495	RTM

4.1.1 Software Unit Description

Additional valid value for DCF line item 4200-050 will be introduced. Value of "N/A" is acceptable for Non-Major Programs; there will be no penalty when this opinion is selected.

4.1.2 Software Unit

Both the internal and external users can access the page by clicking on the Data Collection Form (DCF) link, then selecting "Federal Programs" and then selecting "details".

4.1.3 Accuracy and Validity

If a user selects "No" to DCF line Item "G4200-010", and not selected N/A to DCF line Item"G4200-050, a pop-up warning will alert the user that "G4200-050" must be "N/A" in order to proceed with the DCF. DCF line item "G4200-050" will be set to "N/A" and form field will be disabled (grayed).

Once a user selects other values other than N/A for item line "G4200-010", item line "G4200-050" is set back to editable field.

This function applies to Submission Type "Audited/A-133" only

4.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

4.1.5 Adaptability

This functionality will prevent the user from entering incorrect information for line item G4200-050.

4.2 Environment

4.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

4.2.2 Interfaces

Modification to FASS-PH line item G4200-050 does not interface with other systems.

4.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

4.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

4.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

4.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

4.2.3.4 Temporary and Permanent Storage

There are no new temporary and permanent storage requirements to support this capability.

4.2.4 Security

The following table shows user access rights for reviewing prior years submission comments.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

4.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

4.3 Design Details

The following screenshots are prototypes for DCF line item G4200-050.

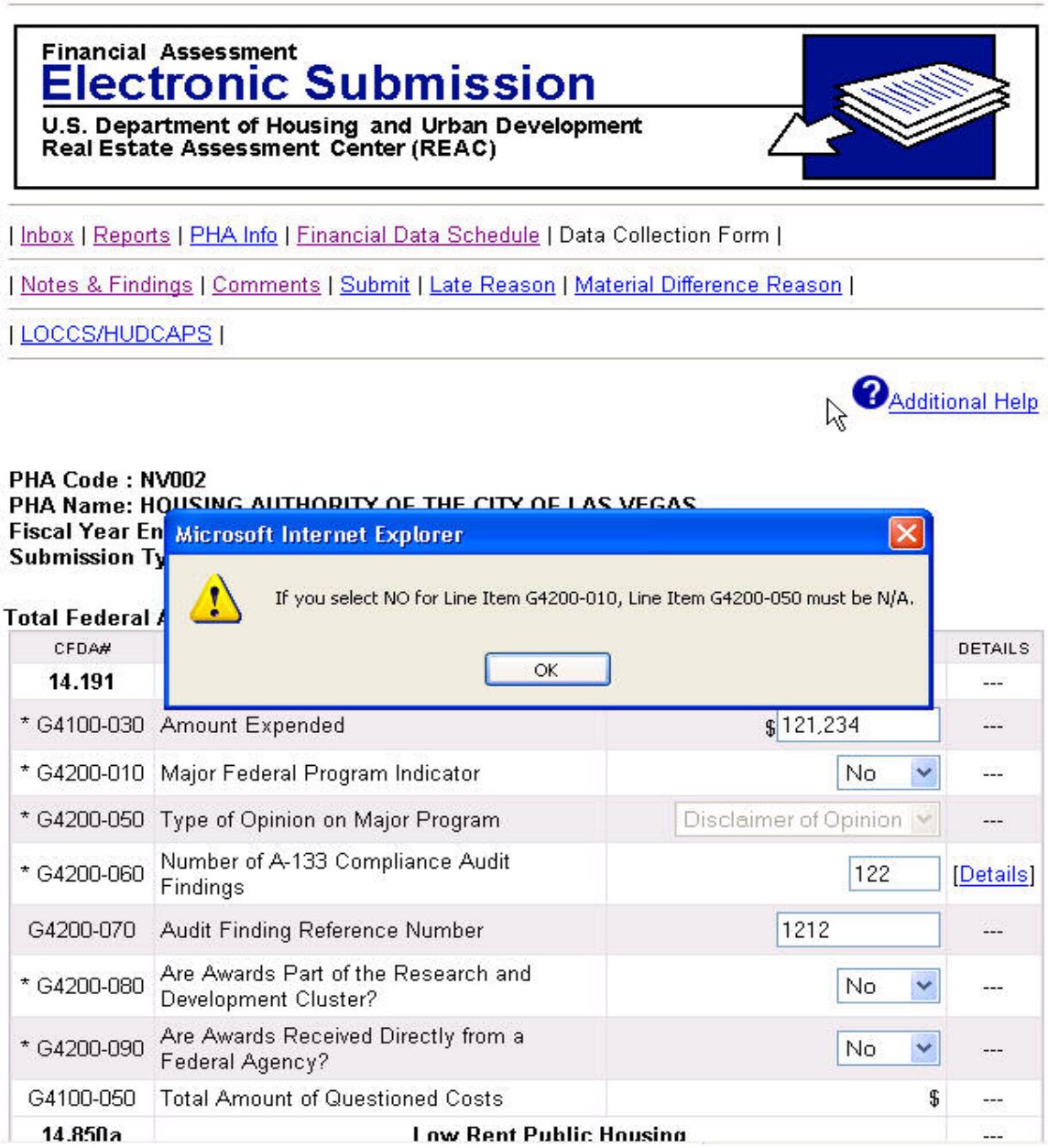


Figure 4-1: Internal and External Users – Line Item G4200-050

Financial Assessment
Electronic Submission
U.S. Department of Housing and Urban Development
Real Estate Assessment Center (REAC)

| Inbox | Reports | PHA Info | Financial Data Schedule | Data Collection Form |

| Notes & Findings | Comments | Submit | Late Reason | Material Difference Reason |

| LOCCS/HUDCAPS |

Additional Help

PHA Code : NV002
PHA Name: HOUSING AUTHORITY OF THE CITY OF LAS VEGAS
Fiscal Year End Date: 09/30/2004
Submission Type: Audited/A-133

Total Federal Awards Expended Details [Back to Federal Programs]

CFDA#	NAME OF FEDERAL PROGRAM		DETAILS
14.191	**Multifamily Housing Service Coordinators**		---
* G4100-030	Amount Expended	$ 121,234	---
* G4200-010	Major Federal Program Indicator	No	---
* G4200-050	Type of Opinion on Major Program	Not Applicable	---
* G4200-060	Number of A-133 Compliance Audit Findings	122	[Details]
G4200-070	Audit Finding Reference Number	1212	---
* G4200-080	Are Awards Part of the Research and Development Cluster?	No	---
* G4200-090	Are Awards Received Directly from a Federal Agency?	No	---
G4100-050	Total Amount of Questioned Costs	$	---
14.850a	**Low Rent Public Housing**		---

Figure 4-2: Internal and External Users – Line Item G4200-050

4.3.1 Input

There are no actual input records.

4.3.1.1 Input Records

There are no actual input records.

4.3.1.2 Input Data Elements

The line items on the database are used as input to the process.

4.3.2 Output

A pop-up warning alert.

4.3.2.1 Output Reports

No reports are generated for this requirement.

4.3.2.2 Output Data Elements

The following table contains the messages that will be displayed if the wrong selection is made.

4.3.3 Software Relationships

No changes for Release 8.1.0.0.

4.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
G4200-010	Yes (REAC)	Submission_line_item	Sub_line_item_ac_smalltext	Varchar(255)	Sub_line_item_ac_smalltext value is 400246	Item_id is 400246
G4200-050	Yes (REAC)	Submission_line_item	Sub_line_item_ac_smalltext	Varchar(255)	Sub_line_item_ac_smalltext value is 401270	Item_id is 401270

Software Unit Logic

Cold Fusion code:

reac_template_dcf_expend_detail_update:

```
<cfif #value# EQ 'Yes'>
<cfset G4200_010_yes_ind = 'Y'>
<cfelseif #value# EQ 'No'>

<cfif NOT IsDefined("form.ac_401270#replacenocase("#trim(program)#",".","")#")>
<cfset sqlaccount[#z#] = "exec faspha_detail_update
#client.participant_id#,#client.group_id#,#client.version_id#,#client.assessment_id#,'#program#',
401270,#client.parent_item_id#,NULL,'Not Applicable','stx',0">
<cfset #z# = #z# + 1>
</cfif>
</cfif>
```

JavaScript code:

reac_template_dcf_expend_detail

```
function check401270(sel, sel401270) {
sel401270.disabled = (sel.options[sel.selectedIndex].value == "No");

if (sel401270.disabled && sel401270.selectedIndex != sel401270.length-1) {
alert('If you select NO for Line Item G4200-010, Line Item G4200-050 must be N/A.');
sel401270.selectedIndex = sel401270.length-1;
                };
        }
```

5.0 LINE ITEM 1102

5.0 LINE ITEM 1102

5.1 Program Description

New logic will be in place, so that the external user will not be able to enter any amount on Line Item G1102. This new methodology should begin for all 9/30/2005 submissions.

New or Modification: Modification

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
1.1,4.1	1.4.1	496	RTM

5.1.1 Software Unit Description

For Fiscal Year 09/30/2005 and beyond allow no entries for Line Item 1102. Whenever the user enters value for this line item, a pop-up message will appear.

5.1.2 Software Unit

Both the internal and external users can access the page by clicking on the Financial Data Schedule (FDS) link, then clicking on the “Revenue & Expense” tab.

5.1.3 Accuracy and Validity

When a user enters a value in line item 1102 starting with Fiscal Year 09/30/2005, a pop-up alert message will appear to alert the user that entering data for the line item is invalid.

5.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

5.1.5 Adaptability

This functionality will prevent users from entering value in line item 1102 starting with Fiscal Year 09/30/2005.

5.2 Environment

5.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.

Software	Description
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

5.2.2 Interfaces

Debt Principal Payments – Line Item 1102 does not interface with other systems.

5.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

5.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

5.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

5.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

5.2.3.4 Temporary and Permanent Storage

There are no new temporary and permanent storage requirements to support this capability.

5.2.4 Security

The following table shows user access rights for reviewing prior years submission comments.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

5.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

5.3 Design Details

The following screenshots are prototypes for Debt Principal payments line item 1102.

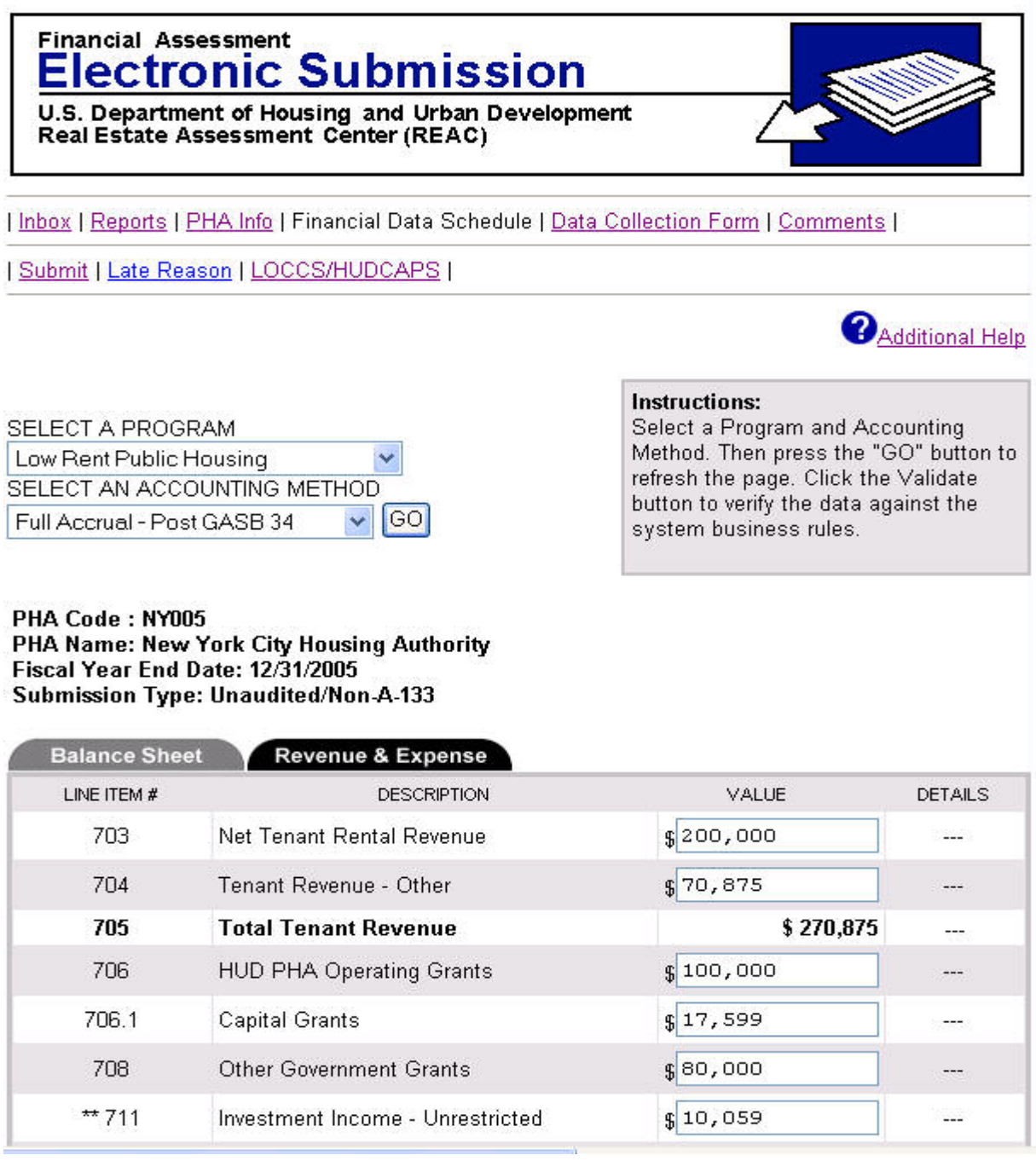


Figure 5-1: Internal and External Users – Financial Data Schedule Revenue & Expense

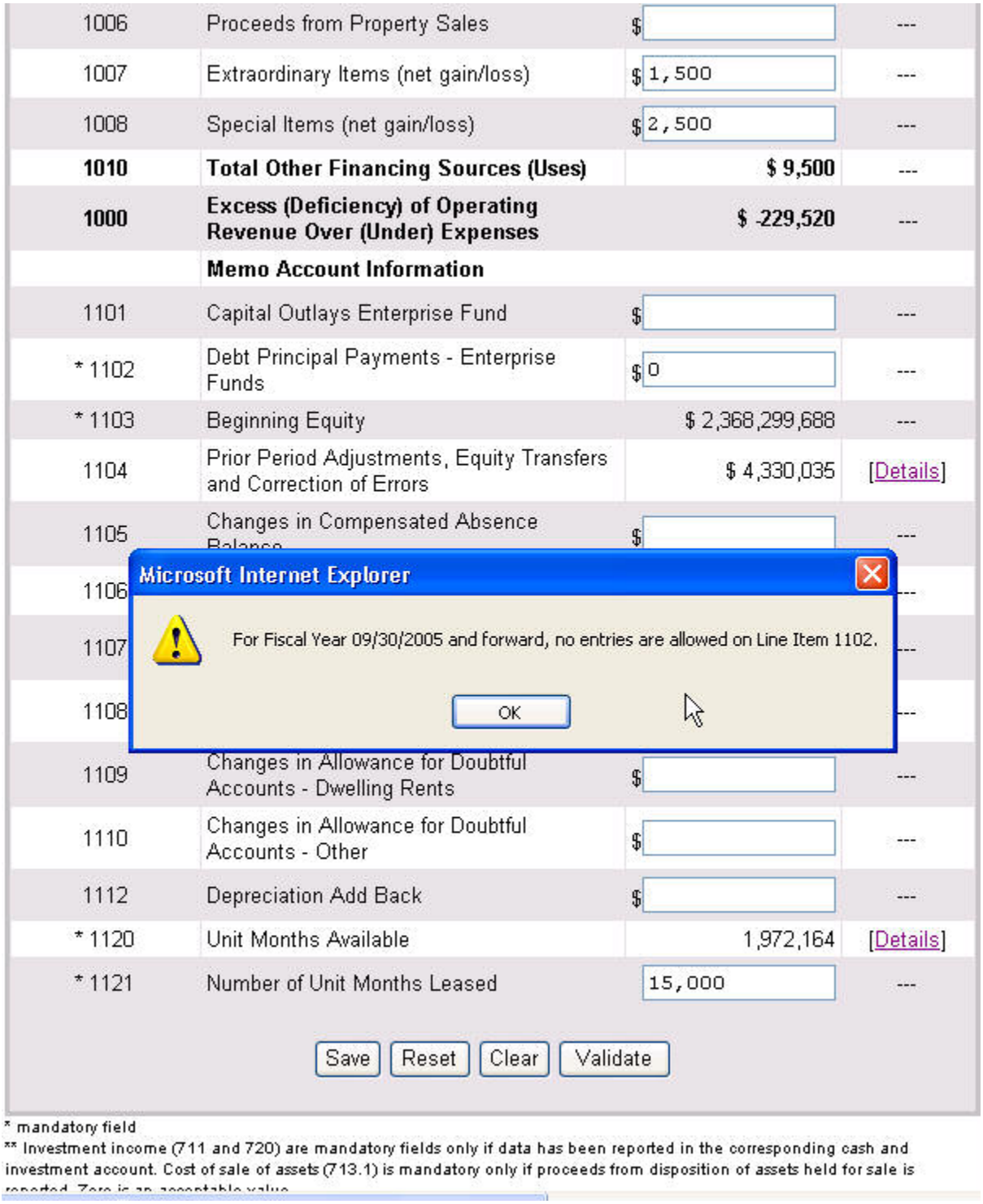


Figure 5-2: Internal and External Users – Financial Data Schedule Revenue & Expense

Financial Data Schedule Revenue & Expense Page

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
1102	Yes (REAC)	Submission_line_item	Sub_line_item_ac_smalltext	Varchar(255)	Sub_line_item_ac_smalltext value is 400004	Item_id is 400004

5.3.1 Input

Input data will be valid for line item 1102 until the end of 09/30/2005 Fiscal Year.

5.3.1.1 Input Records

Input data will be valid for line item 1102 until the end of 09/30/2005 Fiscal Year.

5.3.1.2 Input Data Elements

The line items on the database are used as input to the process.

5.3.2 Output

A pop-up warning alert.

5.3.2.1 Output Reports

No reports are generated for this requirement.

5.3.2.2 Output Data Elements

Line item1102 of Revenue and Expense in the Financial Data Schedule.

5.3.3 Software Relationships

No changes for Release 8.1.0.0.

5.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Financial Data Schedule Revenue & Expense Page

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
1102	Yes (REAC)	Submission_line_item	Sub_line_item_ac_smalltext	Varchar(255)	Sub_line_item_ac_smalltext value is 400004	Item_id is 400004

Software Unit Logic

Cold Fusion codes:

faspha_auto_editflag_selection:

```
<CFIF rule_description EQ '1102'>
<br><FONT FACE="Arial" COLOR="brown" SIZE="-2">
If your submission is for Fiscal Year 9/30/2005 and forward, please ignore this edit flag.
</FONT>
</CFIF>
```

reac_template_fds_data_engine:

```
$<input type="text" size="14" maxlength="14" name="#mainItemList[x].item_html_tag#"

<CFIF sbmt_aftr_05 AND mainItemList[x].item_id IS 400004>
value="0" onFocus="alert('For Fiscal Year 09/30/2005 and forward, no entries are allowed on
Line Item 1102.'); blur()">
<CFELSE>
value="<CFIF
IsNumeric(mainItemList[x].item_value)>#NumberFormat(mainItemList[x].item_value)#</CFIF
>">
</CFIF>
```

6.0 FASS-PH ANALYST INBOX

6.0 FASS ANALYST INBOX

6.1 Program Description

FASS Analyst column of the External Inbox will be modified to display names of the Analyst or Business Manager.

This maintenance requirement will provide the functionality to display only a manager's or financial analyst's name in the FASS Analyst column that last viewed the submission. Following this logic, the director's name should never be display in this column.

New or Modification: Modification

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
5	2.3	498	RTM

6.1.1 Software Unit Description

This maintenance requirement will provide the functionality to display only the manager or financial analyst names in the FASS-PH Analyst column of the External Inbox that last viewed the submission. Following this logic, a director's name is never displayed in the FASS-PH Analyst column.

6.1.2 Software Unit

This modification applies to the external users inbox only. The inbox is accessible once a use logs into the system.

6.1.3 Accuracy and Validity

The system will check role codes and if a director last reviewed a submission, the name of the analyst or business manager that last reviewed the submission will be displayed in the FASS-PH Analyst column.

6.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

6.1.5 Adaptability

Based on the business rule that a Director's name is never displayed in the FASS-PH analyst column, the Analyst's name and manager's name that last review's a submission will be displayed.

6.2 Environment

6.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

6.2.2 Interfaces

FASS-PH Inbox analyst column does not interface with other systems.

6.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

6.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

6.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

6.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

6.2.3.4 Temporary and Permanent Storage

There are no new temporary and permanent storage requirements to support this capability.

6.2.4 Security

The following table shows user access rights for reviewing prior years submission comments.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

6.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

6.3 Design Details

The following screenshots are prototypes for the FASS Analyst Inbox.

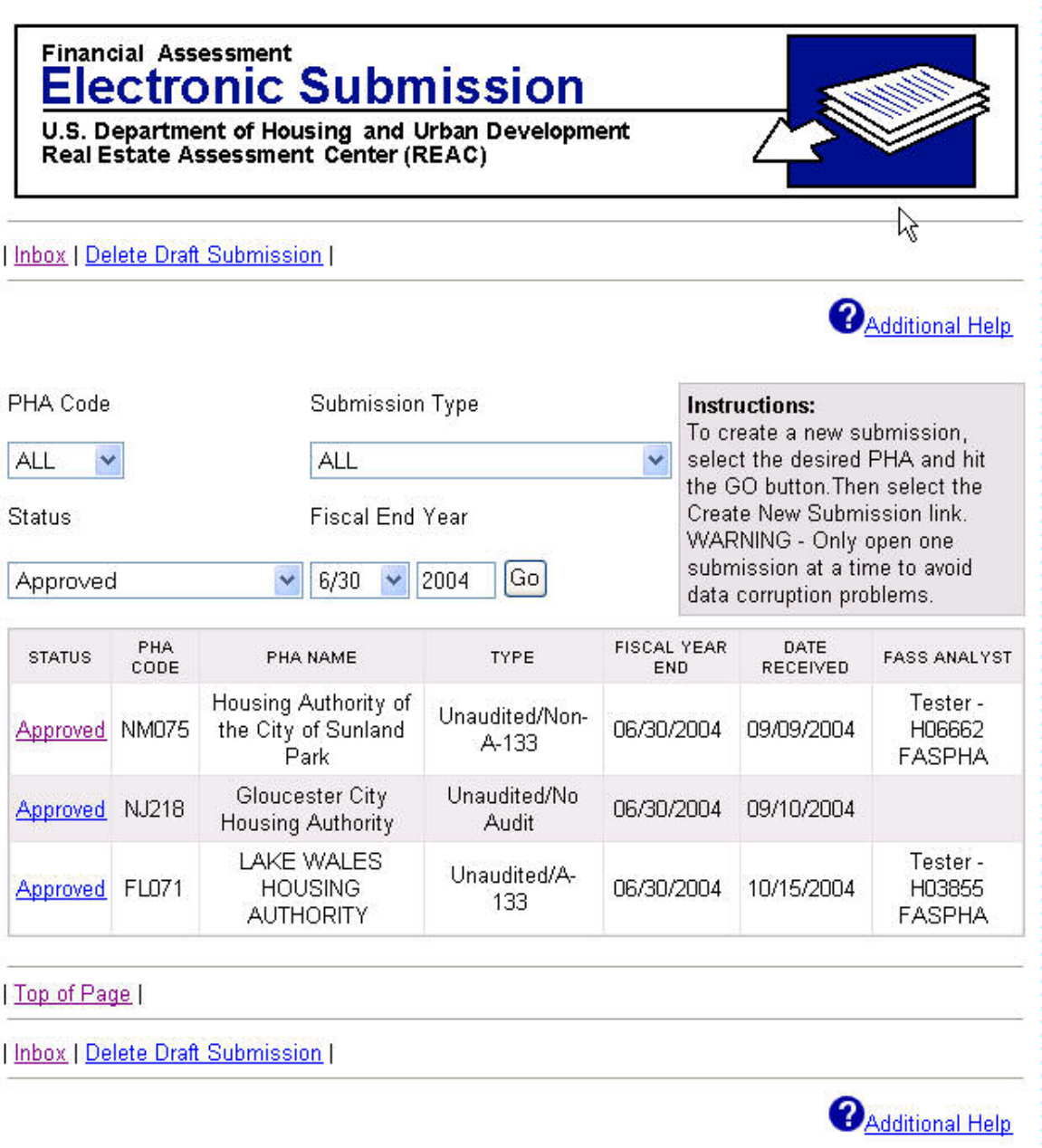


Figure 6-1: External Users Inbox – FASS-PH Analyst Inbox

6.3.1 Input

Input data is user identifier.

6.3.1.1 Input Records

User identifier.

6.3.1.2 Input Data Elements

user_id from the mfs_user table and role_code from the pha_assignment table .

6.3.2 Output

The FASS-PH Analyst column will display the Analyst and Business Manager names.

6.3.2.1 Output Reports

No reports are generated for this requirement.

6.3.2.2 Output Data Elements

user_first_name and user_first_name from the mfs_user table.

6.3.3 Software Relationships

No changes for Release 8.1.0.0.

6.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
FASS Analyst	No (REAC)	mfs_user	user_first_name/ user_last_name	char (20)		FASS Analyst first name and last name

Software Unit Logic

Stored Procedures

faspha_inbox_inter_main:

```
UPDATE #TEMP2
SET role_code = (SELECT a.fk_role_code
                                    FROM pha_assignment a noholdlock
                                    WHERE a.fk_user_id = #TEMP2.analyst_id
                                    AND   a.fk_pih_participant_id = #TEMP2.participant_id
                                    )
WHERE status_name <> 'Draft'
AND analyst_id IS NOT NULL

IF @@error <> 0
        BEGIN
                SELECT @rc = 5, @reason = 'Error setting ROLE Code'
                GOTO COMMON_EXIT
        END
SELECT * INTO #p1 FROM #TEMP2 WHERE role_code = 'FID'
IF @@error <> 0
        BEGIN
                SELECT @rc = 5, @reason = 'Error setting analyst'
                GOTO COMMON_EXIT
        END
        -- Update Analyst ID
UPDATE #p1
SET analyst_id =  (select fk_user_id from pha_assignment p
            where fk_role_code = 'AM'
            AND #p1.participant_id = p.fk_pih_participant_id)
IF @@error <> 0
        BEGIN
                SELECT @rc = 5, @reason = 'Error setting analyst ID in temp table'
                GOTO COMMON_EXIT
        END

UPDATE #TEMP2
SET analyst_id = #p1.analyst_id
FROM #p1
WHERE #p1.participant_id = #TEMP2.participant_id
AND   #p1.assessment_id = #TEMP2.assessment_id
AND   #p1.group_id = #TEMP2.group_id
```

```
AND   #p1.version_id = #TEMP2.version_id

        IF @@error <> 0
        BEGIN
                SELECT @rc = 5, @reason = 'Error setting analyst ID in select  table'
                GOTO COMMON_EXIT
        END


UPDATE #TEMP2
SET fass_analyst = (SELECT a.user_first_name + ' ' + a.user_last_name
                                        FROM mfs_user a noholdlock
                                        WHERE a.user_id = #TEMP2.analyst_id
                                        )
WHERE status_name <> 'Draft'
AND analyst_id IS NOT NULL

IF @@error <> 0
        BEGIN
                SELECT @rc = 5, @reason = 'Error setting analyst'
                GOTO COMMON_EXIT
        END
```

7.0 FDS REPORT PAGE PRINT FUNCTIONALITY

7.0 FDS REPORT PAGE PRINT FUNCTIONALITY

7.1 Program Description

The Financial Data Schedule (FDS) report page breaks wrong printing in Internet Explorer

This maintenance requirement will repair the printing functionality so that page breaks correctly when printing the FDS reports.

New or Modification: Modification

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
6.1	4.1	499	RTM

7.1.1 Software Unit Description

This maintenance requirement will provide the functionality for the FDS report to break correctly when printed from Internet Explorer.

7.1.2 Software Unit

This modification applies to the internal and external systems. The FDS report page print function in Internet Explorer.

7.1.3 Accuracy and Validity

The FDS report page when printed in Internet Explorer will apply page breaks where appropriate.

7.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

7.1.5 Adaptability

Apply page break for FDS report page when printing in Internet Explorer.

7.2 Environment

7.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.

Software	Description
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

7.2.2 Interfaces

The FDS report page print functionality does not interface with other systems.

7.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

7.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

7.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

7.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

7.2.3.4 Temporary and Permanent Storage

There are no new temporary and permanent storage requirements to support this capability.

7.2.4 Security

The following table shows user access rights for reviewing prior years submission comments.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

7.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

7.3 Design Details

The following screenshots are prototypes for the FDS report page.

PHA: **NM075** *FYED:* **06/30/2004**

Line Item No.	Account Description	Low Rent Public Housing	Public Housing Capital Fund Program	Total
111	Cash - Unrestricted	$184,949	$0	$184,949
114	Cash - Tenant Security Deposits	$5,351	$0	$5,351
100	Total Cash	$190,300	$0	$190,300
125	Accounts Receivable - Miscellaneous	$1,495	$0	$1,495
126	Accounts Receivable - Tenants - Dwelling Rents	$630	$0	$630
126.1	Allowance for Doubtful Accounts - Dwelling Rents	$0	$0	$0
126.2	Allowance for Doubtful Accounts - Other	$0	$0	$0
120	Total Receivables, net of allowances for doubtful accounts	$2,125	$0	$2,125
131	Investments - Unrestricted	$180,000	$0	$180,000
143	Inventories	$3,805	$0	$3,805
143.1	Allowance for Obsolete Inventories	$0	$0	$0
150	Total Current Assets	$376,230	$0	$376,230
161	Land	$150,000	$0	$150,000
162	Buildings	$2,492,185	$0	$2,492,185
163	Furniture, Equipment & Machinery - Dwellings	$12,914	$1,098	$14,012
164	Furniture, Equipment & Machinery - Administration	$31,395	$19,912	$51,307
165	Leasehold Improvements	$40,390	$18,908	$59,298
166	Accumulated Depreciation	$-520,539	$0	$-520,539
160	Total Fixed Assets, Net of Accumulated Depreciation	$2,206,345	$39,918	$2,246,263
180	Total Non-Current Assets	$2,206,345	$39,918	$2,246,263
190	Total Assets	$2,582,575	$39,918	$2,622,493

PHA: **NM075** *FYED:* **06/30/2004**

Line Item No.	Account Description	Low Rent Public Housing	Public Housing Capital Fund Program	Total
322	Accrued Compensated Absences - Current Portion	$2,581	$0	$2,581
341	Tenant Security Deposits	$4,950	$0	$4,950
345	Other Current Liabilities	$401	$0	$401
310	Total Current Liabilities	$7,932	$0	$7,932
354	Accrued Compensated Absences - Non Current	$6,510	$0	$6,510
350	Total Noncurrent Liabilities	$6,510	$0	$6,510
300	Total Liabilities	$14,442	$0	$14,442
508	Total Contributed Capital	$0	$0	$0
508.1	Invested in Capital Assets, Net of Related Debt	$2,206,345	$39,918	$2,246,263
511	Total Reserved Fund Balance	$0	$0	$0

Figure 7-1: Internal and External Users Inbox – FDS report page

7.3.1 Input

No input records.

7.3.1.1 Input Records

No input records.

7.3.1.2 Input Data Elements

The line items on the database are used as input to the process.

7.3.2 Output

FDS report.

7.3.2.1 Output Reports

FDS reports.

7.3.2.2 Output Data Elements

The following table contains the messages that will be displayed if the wrong selection is made.

7.3.3 Software Relationships

No changes for Release 8.1.0.0.

7.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
N/A	N/A	N/A	N/A	N/A	N/A	N/A

Software Unit Logic

Stored Procedures

N/A

Cold Fusion code:

reac1_FDSPrint:

<CFSET END_LOOP_COUNT = (50 - #TOTAL_ROW_ITEMS#)/4 - 3>

8.0 FILE ATTACHMENTS STORAGE

8.0 FILE ATTACHMENTS STORAGE

8.1 Program Description

All file attachments in FASS-PH assessments must now be stored as Binary Large Objects (BLOBs) in the database as mandated by the Development Coordination Group (DCG).

This infrastructure requirement will replace the current method of storing file attachments in directories on the server. The new method will allow file attachments to be stored as BLOBs in the database.

New or Modification: New

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
	1.2	501	RTM

8.1.1 Software Unit Description

This infrastructure requirement will provide the functionality of storing all file attachments submitted through the FASS-PH system as BLOBs in the REAC database. The current method of storing file attachments in directories on the server will no longer be available as of the 8.1.0.0 Release.

8.1.2 Software Unit

This infrastructure requirement applies to FASS-PH internal and external systems. Storage method for all file attachments accompanying a submission will be changed. File attachments will now have to be stored as BLOBs in the database. PDUMP and iplappsp1 file storage systems will be removed. A database table in the REAC database has been created for storing of the attachments. Cold Fusion pages will be created for uploading and downloading file attachments from the database.

8.1.3 Accuracy and Validity

This functionality is entirely on the backend and will not impact the interface or function of the system.

8.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

8.1.5 Adaptability

Apply page break for FDS report page when printing in Internet Explorer.

8.2 Environment

8.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

8.2.2 Interfaces

The FDS report page print functionality does not interface with other systems.

8.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

8.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

8.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

8.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

8.2.3.4 Temporary and Permanent Storage

New table was created in the REAC database for storage of file attachments.

8.2.4 Security

The following table shows user access rights.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

8.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

8.3 Design Details

N/A.

8.3.1 Input

File attachments submitted with financial submissions.

8.3.1.1 Input Records

File attachments will be input into the submission_line_item_doc table.

8.3.1.2 Input Data Elements

The line items on the database are used as input to the process.

8.3.2 Output

File attachments download.

8.3.2.1 Output Reports

File attachments download.

8.3.2.2 Output Data Elements

The file_content data element in the submission_line_item_doc table of the REAC database outputs content of the selected file attachment.

8.3.3 Software Relationships

No changes for Release 8.1.0.0.

8.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
N/A	No (REAC)	submission_line_item_doc	participant_id	numeric (10,0)	participant_id	It contains participant_id
N/A	No (REAC)	submission_line_item_doc	group_id	smallint	group_id	It contains group_id
N/A	No (REAC)	submission_line_item_doc	version_id	smallint	version_id	It contains version_id
N/A	No (REAC)	submission_line_item_doc	assessment_id	numeric (9,0)	assessment_id	It contains assessment_id

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
N/A	No (REAC)	submission_line_item_doc	sub_line_item_id	numeric (11,0)	sub_line_item_id	It contains sub_line_item_id
N/A	No (REAC)	submission_line_item_doc	path_name	varchar (255)	path_name	It contains path_name
N/A	No (REAC)	submission_line_item_doc	file_name	varchar (255)	file_name	It contains file_name
N/A	No (REAC)	submission_line_item_doc	file_type	varchar (255)	file_type	It contains file_type

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
N/A	No (REAC)	submission_line_item_doc	file_description	varchar (255)	file_description	It contains file_description
N/A	No (REAC)	submission_line_item_doc	file_virus_scanned	char (1)	file_virus_scanned	It contains file_virus_scanned
N/A	No (REAC)	submission_line_item_doc	file_content	image	file_content	It contains file_content
N/A	No (REAC)	submission_line_item_doc	last_update_date	datetime	last_update_date	It contains last_update_date

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
N/A	No (REAC)	submission_line_item_doc	last_update_user_id	char (6)	last_update_user_id	It contains last_update_user_id

Software Unit Logic

Stored Procedures

faspha_insert_sub_line_item:

```
@participant_id                 NUMERIC(10,0),
@assessment_id                  NUMERIC(9,0),
@group_id                       SMALLINT,
@version_id                     SMALLINT,
@item_id                              INT,
@filename
                                VARCHAR(255)
AS
DECLARE @rc INT
DECLARE @sub_line_item_id numeric(11,0)
 SELECT @sub_line_item_id = -1
SELECT @rc = 0

SELECT @sub_line_item_id = ISNULL(MAX(sub_line_item_id), 0)+1
FROM submission_line_item NOHOLDLOCK
WHERE participant_id = @participant_id
and assessment_id = @assessment_id
and group_id = @group_id
and version_id = @version_id

IF @@error <> 0
BEGIN
SELECT @rc = 0
GOTO COMMON_EXIT
END

if @@trancount = 0
set chained off

BEGIN TRANSACTION
INSERT into submission_line_item
(participant_id, group_id, version_id, assessment_id, sub_line_item_id, program_participant_id,
program_group_id, program_version_id, program_assessment_id, program_ref_id, item_id,
item_group_id ,item_version_id, threshold_item_id, threshold_group_id,
threshold_group_version_id, threshold_range_id,  sub_line_item_ac_value,
sub_line_item_ac_text,sub_line_item_ac_smalltext, sub_line_item_ac_decimal,
sub_line_item_ac_date, accounting_method_ref_id,  project_id, pgm_proj_participant_id,
```

```
pgm_proj_group_id, pgm_proj_version_id, pgm_proj_assessment_id,
pgm_proj_program_ref_id)
VALUES (@participant_id,
        @group_id,
        @version_id,
        @assessment_id,
        @sub_line_item_id,
        @participant_id,
        @group_id,
        @version_id,
        @assessment_id,
        NULL,
        @item_id,
        @group_id,
        @version_id,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        @filename,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL)

        IF @@error <> 0
        BEGIN
        SELECT @rc = -2
        ROLLBACK
                END
        COMMIT

COMMON_EXIT:
        SELECT @sub_line_item_id as sub_line_item_id
        RETURN @rc
GO
IF EXISTS (SELECT * FROM sysobjects WHERE name ='faspha_insert_sub_line_item' and
type='P')
```

```
  BEGIN
        PRINT 'STORED PROCEDURE : faspha_insert_sub_line_item - successfully added'
  END
ELSE
  BEGIN
        PRINT 'STORED PROCEDURE : faspha_insert_sub_line_item - NOT added'
  END

GO
```

Cold Fusion codes:

1. **f_notes_findings:**

```
<CFIF #SUBLINEITEMQUERY.SUB_LINE_ITEM_AC_TEXT# NEQ "">
<INPUT CLASS="BUTTON" TYPE="BUTTON" NAME="attachFile"
VALUE="Attach File"onClick="alert('Before uploading new attachment file.\nDelete the
old file first.')">
<INPUT CLASS="BUTTON" TYPE="submit" NAME="deleteFile" VALUE="Delete
File"onClick="return confirm('Do you really want to delete uploaded file?\nIf yes, click
OK.')">
<CFELSE>
<INPUT CLASS="BUTTON" TYPE="submit" NAME="attachFile" VALUE="Attach
File" onclick="return openpop()">
</CFIF>
</FONT>
</TD>
<CFELSE>

<TD ALIGN="CENTER">
<FONT FACE="Arial, Helvetica, sans-serif" SIZE="-1">
                                                    ---
</FONT>
</TD>
</CFIF>

<CFIF #SUBLINEITEMQUERY.SUB_LINE_ITEM_AC_TEXT# NEQ "">

<TD ALIGN=CENTER VALIGN="middle">
<FONT FACE=ARIAL SIZE=-1>
<CFQUERY NAME="sublineitemdoc" DATASOURCE="#app_datasource#">
select * from submission_line_item_doc
                                                    where participant_id =
    #CLIENT.participant_id#
                                                    and group_id = #CLIENT.group_id#
```

```
and version_id = #CLIENT.version_id#
and assessment_id = #CLIENT.assessment_id#
and sub_line_item_id = #SUBLINEITEMQUERY.sub_line_item_id#
</CFQUERY>
<CFOUTPUT>
<CFIF sublineitemdoc.recordcount EQ 1>
<a href=
"/DCGtools/FileDownload?SystemName=faspha&participant_id=#CLIENT.participant_id#&gr
oup_id=#CLIENT.group_id#&version_id=#CLIENT.version_id#&assessment_id=#CLIENT.ass
essment_id#&sub_line_item_id=#SUBLINEITEMQUERY.sub_line_item_id#">Open File</a>

<CFELSE>
(not available)
</CFIF>
</CFOUTPUT>
```

2. f_notes_findings_action:

```
<CFTRANSACTION>
<CFQUERY NAME="del" DATASOURCE="#app_datasource#">
DELETE FROM submission_line_item_doc
WHERE participant_id = #CLIENT.participant_id#
AND group_id = #CLIENT.group_id#
AND version_id = #CLIENT.version_id#
AND assessment_id = #CLIENT.assessment_id#
AND sub_line_item_id = (SELECT sub_line_item_id
FROM submission_line_item
WHERE participant_id = #CLIENT.participant_id#
AND group_id = #CLIENT.group_id#
AND version_id = #CLIENT.version_id#
AND assessment_id = #CLIENT.assessment_id#
AND item_id = #itemquery.item_id#)
</CFQUERY>
<CFQUERY NAME="del1" DATASOURCE="#app_datasource#">
DELETE FROM submission_line_item
WHERE participant_id = #CLIENT.participant_id#
AND group_id = #CLIENT.group_id#
AND version_id = #CLIENT.version_id#
AND assessment_id = #CLIENT.assessment_id#
AND item_id = #itemquery.item_id#
</CFQUERY>
</CFTRANSACTION>
<CFLOCATION
url="f_notes_findings.cfm?form_ref=#form_ref#&UploadSuccess=D">
    </CFIF>
```

3. reac_submission_comments:

```
function  openpop() {

window.open('faspha_app_upload_servlet.cfm','addAttachment','status,name=teamAp
p,resizable=1,top=250,left=250,width=550,height=135');
    }

<CFSET CLIENT.item_id = newitemquery.item_id>


<CFIF #SUBLINEITEMQUERY.SUB_LINE_ITEM_AC_TEXT# NEQ "">
<INPUT CLASS="BUTTON" TYPE="BUTTON" NAME="attachFile"
VALUE="Attach File" onClick="alert('Before uploading new attachment
file.\nDeletethe old file first.')">

<INPUT CLASS="BUTTON" TYPE="submit" NAME="deleteFile"
VALUE="Delete File" onClick="return confirm('Do you really want to delete
uploaded file?\nIf yes, click OK.')">
<CFELSE>
<INPUT CLASS="BUTTON" TYPE="BUTTON" NAME="attachFile"
VALUE="Attach File" onclick="openpop()">
</CFIF>

<CFOUTPUT>
<a href=
"/DCGtools/FileDownload?SystemName=faspha&participant_id=#phaquery.particip
ant_id#&group_id=#phaquery.group_id#&version_id=#phaquery.version_id#&asses
sment_id=#phaquery.assessment_id#&item_id=#subLineItemquery.item_id#"
target="_blank">Open File</a>
</CFOUTPUT>
```

4. reac_submission-comments_action:

```
CFIF ISDEFINED("FORM.deleteFile")>
    <!--- Do delete --->
    <CFQUERY NAME="del" DATASOURCE="#app_datasource#">
            DELETE FROM submission_line_item_doc
            WHERE participant_id = #CLIENT.participant_id#
            AND group_id = #CLIENT.group_id#
            AND version_id = #CLIENT.version_id#
            AND assessment_id = #CLIENT.assessment_id#
            AND sub_line_item_id = (SELECT sub_line_item_id
                        FROM submission_line_item
```

```
                    WHERE participant_id = #CLIENT.participant_id#
                    AND group_id = #CLIENT.group_id#
                    AND version_id = #CLIENT.version_id#
                    AND assessment_id = #CLIENT.assessment_id#
                    AND item_id = #CLIENT.item_id#)

            DELETE FROM submission_line_item
            WHERE participant_id = #CLIENT.participant_id#
            AND group_id = #CLIENT.group_id#
            AND version_id = #CLIENT.version_id#
            AND assessment_id = #CLIENT.assessment_id#
            AND item_id = #CLIENT.item_id#
        </CFQUERY>

    </CFIF>

    <CFLOCATION URL="reac_submission_comments.cfm" ADDTOKEN="No">
```

5. faspha_app_upload_servlet:

9.0 RELATIVE SERVER NAME

9.0 REALTIVE SERVER NAME

9.1 Program Description

All hard coded port number will be removed from all Cold Fusion templates and replaced with the relative server name.

New or Modification: New

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
	1	505	RTM

9.1.1 Software Unit Description

This infrastructure requirement will change the use of hard coded https port 801 number to relative server name on the production fix server.

9.1.2 Software Unit

Production fix server use of hard coded port number modified to use the relative server name.

9.1.3 Accuracy and Validity

This functionality is entirely on the backend and will not impact the interface or function of the system.

9.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

9.1.5 Adaptability

Change all hard coded port number to relative server name.

9.2 Environment

9.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

9.2.2 Interfaces

N/A.

9.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

9.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

9.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

9.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

9.2.3.4 Temporary and Permanent Storage

There are no temporary and permanent storage requirements to support this capability.

9.2.4 Security

The following table shows user access rights.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

9.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

9.3 Design Details

N/A.

9.3.1 Input

N/A.

9.3.1.1 Input Records

N/A.

9.3.1.2 Input Data Elements

N/A.

9.3.2 Output

N/A.

9.3.2.1 Output Reports

N/A.

9.3.2.2 Output Data Elements

N/A.

9.3.3 Software Relationships

No changes for Release 8.1.0.0.

9.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
N/A	N/A	N/A	N/A	N/A	N/A	N/A

Software Unit Logic

Cold Fusion application file was modified to remove all instances of hard coded port number on the Production Fix server.

Cold Fusion code:

application.cfm

```
<CFELSEIF  FIND("/opt/suitespot/prod", ServerPath,1) GT 0>
        <CFSET APP_DATASOURCE = "p08a_reacs">
        <CFSET APP_SECURITY_DATA = "p08a_herems">
        <CFSET APP_WASS_DATASOURCE = "p08a_p104">
        <CFSET APP_FASPHA_EXTERNAL_ROOT =
"https://#URLHOST#/HUD_Systems/faspha">
        <CFSET APP_FASPHA_INTERNAL_ROOT =
"https://#URLHOST#/ssmaster/faspha">
        <CFSET APP_QASS_INTERNAL_ROOT = "/ssmaster/qass">
        <CFSET VARIABLE.INTERNAL_SERVER = "/ssmaster/">
        <CFSET VARIABLE.EXTERNAL_SERVER = "/HUD_Systems/">
```

Stored Procedures

N/A.

10.0 CHECK BOX FOR GUEST USERS

10.0 CHECK BOX FOR GUEST USERS

10.1 Program Description

This requirement calls for modification to how a guest user logs into FASS-PH. Currently a guest user manually checks the Guest User checkbox on the login interface. Web Access Subsystem (WASS) will remove the Guest user checkbox beginning with Release 8.1.0.0. The FASS-PH system will implement a new functionality on how a guest user is identified. The system will search to find if any Role ID's is associated with the User ID. If a Role ID's is associated with the User ID, the user will be redirected to the Main Internal page. If no Role ID is associated with the User ID, the user will be redirected to the Main Guest Page

New or Modification: Modification.

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
1.1	1	497	RTM

10.1.1 Software Unit Description

The requirement to remove the Check Box for guest users will provide a modified process of logging guest users into the FASS PH. Currently a guest user manually checks the Guest User checkbox on the login interface. Web Access Security System (WASS) will remove the CHECK BOX from the logon screen. When a user logs on to WASS and select FASS-PH, WASS will send a guest role code for the user. FASS-PH will check its role table for that user; if the user has another role code then he will be assigned his FASS-PH role code. If the user does not have a FASS-PH role code, then his role code will default to GUEST.

10.1.2 Software Unit

This page is accessible to internal users only. The guest users check box, when checked, will direct the user to a guest page. The guest page does not contain links and other screens that are accessible to users with other roles. The new functionality will check for guest users and other role codes based on the user identification entered by the user.

10.1.3 Accuracy and Validity

All users that login to the FASS-PH internal system have a guest role. FASS-PH users have other roles besides that of a guest. Beginning with Release 8.1.0.0, when a user logs into the FASS-PH through the WASS interface, the system will check for role codes for that user, if the user has other role codes besides that of a guest, the user is directed to the main internal page, but if the user has only the role of a guest, the user is directed to the main guest page.

10.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

10.1.4.1 Adaptability

Removal of check box from WASS internal login interface is an infrastructure requirement that has to be adapted by FASS-PH. WASS controls access to the FASS-PH subsystem.

10.2 Environment

10.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

10.2.2 Interfaces

The FASS-PH internal system interfaces with WASS for this requirement.

10.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

10.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

10.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

10.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

10.2.3.4 Temporary and Permanent Storage

There are no temporary and permanent storage requirements to support this capability.

10.2.4 Security

The following table shows user access rights.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

10.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

10.2 Design Details

The following screenshots is a prototype of the Check Box for Guest Users on the WASS interface.

Figure 10-1: WASS Internal System Interface.

10.3.1 Input

User Identifier on the WASS login interface

10.3.1.1 Input Records

Username and Password.

10.3.1.2 Input Data Elements

user_id and user_passwd.

10.3.2 Output

Main internal page or main guest page is displayed based on user's role code.

10.3.2.1 Output Reports

N/A.

10.3.2.2 Output Data Elements

N/A.

10.3.3 Software Relationships

No changes for Release 8.1.0.0.

10.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
N/A	N/A	N/A		N/A	N/A	N/A

Software Unit Logic

Check for user role codes, if user has other role codes besides guest, redirect to the main internal inbox. If user has only guest role, redirect to the main guest page.

Stored Procedures

N/A

Cold Fusion code:

reac_inbox_intra_first:

```
<cfif (#ListFindNoCase(#rolelist#,"GHU",",")# EQ 0) OR #rolecode.recordcount# GT 1>
  <cflocation url="reac_inbox_intra_bridge.cfm?RequestTimeout=300" addtoken="No">
<cfelse>
     <CFSET client.user_id = "guest">
  <cflocation url="reac_inbox_intra_guest_state.cfm" addtoken="No">
</cfif>
```

11.0 IDENTITY TYPE

11.0 IDENTITY TYPE

11.1 Program Description

The participant_assessment table contains a column named assessment_id, which has a column definition of IDENTITY. The column definition allows the assessment_id to automatically increment its value by 1 whenever a new row is added to the table. The removal of the IDENTITY column definition from the REAC database by the Development Coordination Group (DCG) calls for the FASS-PH system to be changed to get the MAX of the ASSESSMENT_ID column increment that value by 1 when adding a new row to the table.

New or Modification: Modification.

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
	1.2	503	RTM

11.1.1 Software Unit Description

The participant_assessment table contains a column named assessment_id, which has a column definition of IDENTITY. The column definition allows the assessment_id to automatically increment its value by 1 whenever a new row is added to the table. The removal of the IDENTITY column definition from the REAC database by the Development Coordination Group (DCG) calls for the FASS-PH system to be changed to get the MAX of the ASSESSMENT_ID column increment that value by 1 when adding a new row to the table.

11.1.2 Software Unit

This requirement applies to the internal FASS-PH system users that need to create submissions. External users can create submission only if they have ‘SMT’ role code. Users will get to the PHA INFO Page by clicking on the “PHA INFO” tab.

11.1.3 Accuracy and Validity

Whenever a new row is added to the participant_assessment table, the assessment_id column will increment by the value of one.

11.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

11.1.5 Adaptability

DCG is removing the IDENTITY column definition from the participant_assessment table in the REAC database and this calls for FASS-PH to create a new functionality of incrementing new rows added to the participant_assessment table.

11.2 Environment

11.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

11.2.2 Interfaces

The participant_assessment table is in the REAC database. The REAC database is utilized by all the subsystems.

11.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

11.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

11.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

11.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

11.2.3.4 Temporary and Permanent Storage

There are no temporary and permanent storage requirements to support this capability.

11.2.4 Security

The following table shows user access rights.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

11.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

11.3 Design Details

The following screenshots is a prototype of the PHA Info page.

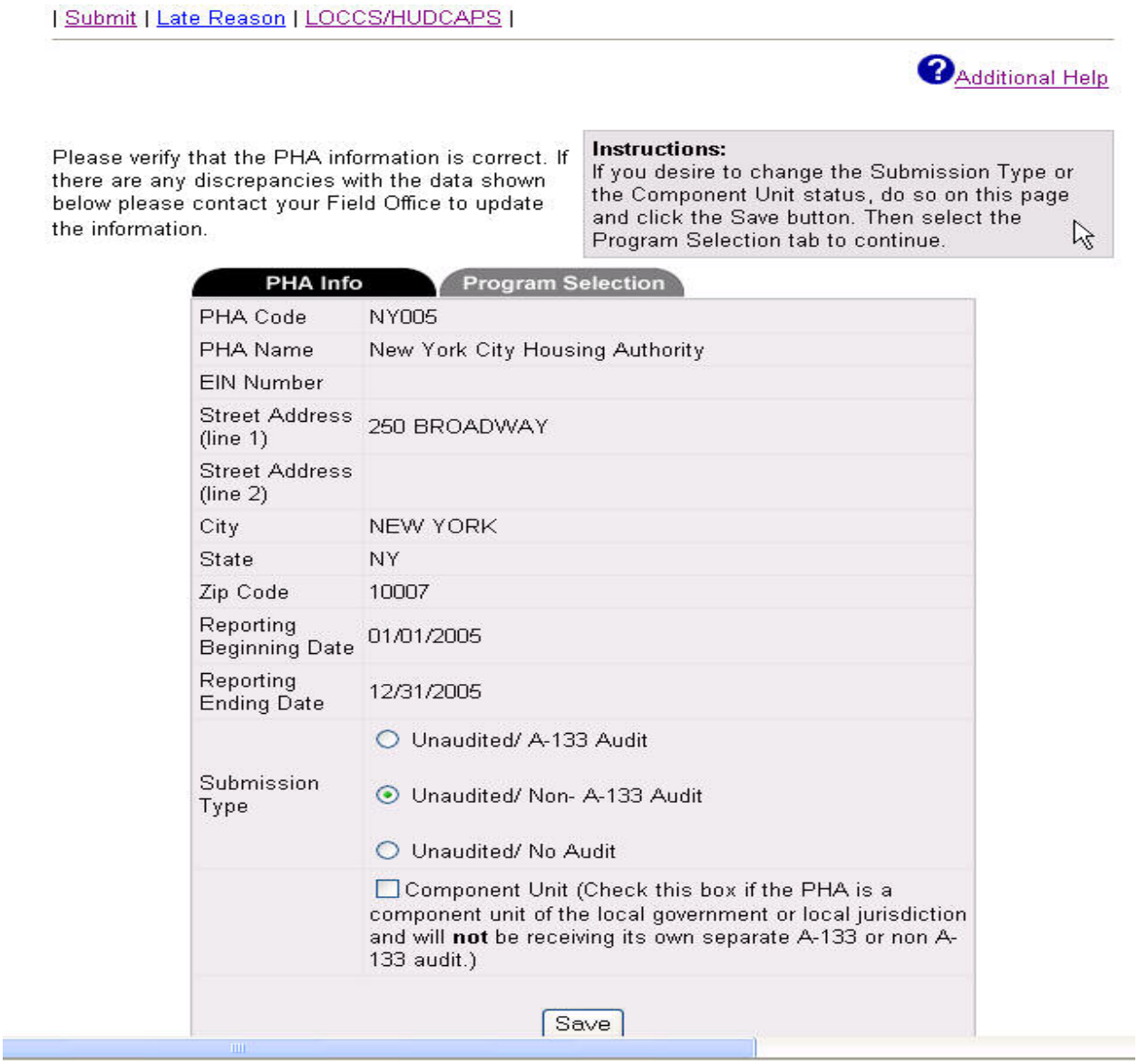


Figure 11.1: PHA Info Screenshot

11.3.1 Input

participant_id
group_id
version_id
fiscal_year
submission_type
participant_yend
user_id
intranet
component_unit
participant_rpt_to_date
participant_rpt_from_date
group_id smallint,
version_id smallint,
assessment_id_old
fiscal_year
newsub_group_id
component_unit_ind

11.3.1.1 Input Records

New submissions

11.3.1.2 Input Data Elements

assessment_id

11.3.2 Output

N/A.

11.3.2.1 Output Reports

N/A.

11.3.2.2 Output Data Elements

N/A.

11.3.3 Software Relationships

No changes for Release 8.1.0.0.

11.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
Identity	No (REAC)	Participant_assessment	Assessment_id	Numeric(9)	Assessment_id	Identity of Assessment

Software Unit Logic

Stored Procedures

reac_faspha_new_submission:

```
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'reac_faspha_new_submission')
  BEGIN
    PRINT 'DROP PROCEDURE reac_faspha_new_submission'
    DROP PROCEDURE reac_faspha_new_submission
  END
GO


CREATE PROCEDURE reac_faspha_new_submission

 @participant_id numeric(10),
 @group_id smallint,
 @version_id smallint,
 @fiscal_year numeric(4),
 @submission_type varchar(20),
 @participant_yend datetime,
 @user_id char(6),
 @intranet char(1),
 @component_unit char(1),
 @participant_rpt_to_date datetime,
 @participant_rpt_from_date datetime = NULL,
 @assessment_id        numeric(9) OUTPUT

AS

BEGIN tran

        DECLARE @assessment                numeric(9),
                @assess_action_id          numeric(10),
                @object_assignment_id      smallint,
                @phacode                   char(5),
                @phaname                   varchar(60) ,
                @object_id                 smallint,
                @action_code               char(3),
                @system_id                 char(6),
                @rc                        integer,
```

```
                @reason                 varchar(255)


  SELECT @rc = 0, @reason = NULL, @system_id = 'FASPHA'

  SELECT        @phacode=pha_id,
                @phaname=participant_common_name
  FROM          pha_participant noholdlock
  WHERE         participant_id=@participant_id
  IF @@error <> 0 GOTO SQL_ERROR

  exec dcg_return_key 'asmtid', 9, @assessment output
   IF @@error <> 0 GOTO SQL_ERROR

   SELECT @assessment_id = @assessment
   IF @@error <> 0 GOTO SQL_ERROR

  INSERT INTO participant_assessment
  (
participant_id,group_id,version_id,assessment_id,effective_date,assessment_score,
  fiscal_year,status_code,risk_id,submission_type,pha_id,
  participant_fiscal_yr_end_date,participant_common_name,component_unit_ind,
  reporting_period_to_dt,reporting_period_from_dt
  )
  VALUES
  (
@participant_id,@group_id,@version_id,@assessment,@participant_yend,NULL,
  @fiscal_year,NULL,NULL,@submission_type,@phacode,
  @participant_yend,@phaname,@component_unit,@participant_rpt_to_date,
  @participant_rpt_from_date
  )
  IF @@error <> 0 GOTO SQL_ERROR

  IF UPPER(@intranet) = 'N'
  BEGIN

          INSERT INTO participant_assess_action
          (
          object_id,action_code,system_id,
          participant_id,group_id,version_id,assessment_id,
          action_date,user_id,
        action_amount,action_comment,board_resolution_number,certification_date
          )
          VALUES
          (
```

```
        1,'CSB','FASPHA',
        @participant_id,@group_id,@version_id,@assessment,
        getdate(),@user_id,
        NULL,NULL,NULL,NULL
        )
        IF @@error <> 0 GOTO SQL_ERROR

        END
        ELSE
        BEGIN
        INSERT INTO participant_assess_action
        (
        object_id,action_code,system_id,
        participant_id,group_id,version_id,assessment_id,
        action_date,user_id,

action_amount,action_comment,board_resolution_number,certification_date
        )
        VALUES
        (
        5,'CRM','FASPHA',
        @participant_id,@group_id,@version_id,@assessment,
        getdate(),@user_id,
        NULL,NULL,NULL,NULL
        )
        IF @@error <> 0 GOTO SQL_ERROR
END

SELECT @assess_action_id=@@identity

        SELECT @object_id = object_id, @action_code = action_code,
                @system_id = system_id
        FROM participant_assess_action
        WHERE participant_id = @participant_id
        AND group_id = @group_id
        AND version_id = @version_id
        AND assessment_id = @assessment
        AND participant_assess_action_id = @assess_action_id
IF @@error <> 0 GOTO SQL_ERROR

SELECT @object_assignment_id=MAX(object_assignment_id)
        FROM object_assignment_queue
        WHERE object_id = @object_id
        AND action_code = '@action_code'
        AND system_id = @system_id
```

```
            AND participant_assess_action_id = @assess_action_id
        IF @@error <> 0 GOTO SQL_ERROR

        IF @object_assignment_id = NULL
            SELECT @object_assignment_id=1
        ELSE
            SELECT @object_assignment_id=@object_assignment_id + 1

        IF UPPER(@intranet) = 'N'
        BEGIN
            INSERT INTO object_assignment_queue
            (object_id,action_code,system_id,participant_assess_action_id,
            object_assignment_id,
            in_queue_dt,out_queue_dt,userid,assigning_userid)
            VALUES
            (1,'CSB','FASPHA',@assess_action_id,@object_assignment_id,
            getdate(),NULL,@user_id,@user_id)
            IF @@error <> 0 GOTO SQL_ERROR
        END
        ELSE
        BEGIN
            INSERT INTO object_assignment_queue
            (object_id,action_code,system_id,participant_assess_action_id,
            object_assignment_id,in_queue_dt,out_queue_dt,userid,assigning_userid)
            VALUES
            (5,'CRM','FASPHA',@assess_action_id,@object_assignment_id,
            getdate(),NULL,@user_id,@user_id)
            IF @@error <> 0 GOTO SQL_ERROR
        END

        GOTO SQL_ERROR

SQL_ERROR:

IF @@error > 0
        BEGIN
            SELECT @rc = 12, @reason = 'Failure in SQL Statement'
            SELECT @rc AS RC, @reason AS REASON
            RETURN- 12
        END
ELSE
        BEGIN
            COMMIT tran
            SELECT @rc AS RC, @reason AS REASON
            RETURN 0
```

```
        END

GO
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'reac_faspha_new_submission')
  BEGIN
     PRINT 'CREATE PROCEDURE reac_faspha_new_submission SUCCESSFUL'
     EXEC sp_help reac_faspha_new_submission
  END
ELSE
  BEGIN
    PRINT 'CREATE PROCEDURE reac_faspha_new_submission FAILED'
  END
GO


IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'reac_faspha_download_sub')
  BEGIN
     PRINT 'DROP PROCEDURE reac_faspha_download_sub'
     DROP PROCEDURE reac_faspha_download_sub
  END
GO
```

stored procedure to download file:

reac_faspha_download_sub:

```
CREATE PROCEDURE reac_faspha_download_sub

 @participant_id numeric(10),
 @group_id smallint,
 @version_id smallint,
 @assessment_id_old numeric(9),
 @fiscal_year numeric(4),
 @submission_type varchar(20),
 @participant_yend datetime,
 @user_id char(6),
 @intranet char(1),
 @newsub_group_id smallint,
 @component_unit_ind char(1),
 @participant_rpt_to_date datetime,
 @participant_rpt_from_date datetime = NULL,
 @assessment_id        numeric(9) OUTPUT
```

```
AS

BEGIN TRAN

        DECLARE     @assessment             numeric(9),
                    @assess_action_id       numeric(10),
                    @object_assignment_id   smallint,
                    @phacode                char(5),
                    @phaname                varchar(60),
                    @object_id              smallint,
                    @action_code            char(3),
                    @system_id              char(6),
                    @rc                     integer,
                    @reason                 varchar(255)

        select @rc = 0, @reason = NULL, @system_id = 'FASPHA'

        SELECT @phacode=pha_id,@phaname=participant_common_name
                FROM pha_participant noholdlock
                WHERE participant_id=@participant_id
        IF @@error <> 0 GOTO SQL_ERROR

         exec dcg_return_key 'asmtid', 9, @assessment output
         IF @@error <> 0 GOTO SQL_ERROR

         SELECT @assessment_id = @assessment
         IF @@error <> 0 GOTO SQL_ERROR

         INSERT INTO participant_assessment
         (
        participant_id,group_id,version_id,assessment_id,effective_date,
        assessment_score, fiscal_year,status_code,risk_id,submission_type,pha_id,
        participant_fiscal_yr_end_date,participant_common_name,component_unit_ind,
         reporting_period_to_dt,reporting_period_from_dt
        )
        VALUES
        (
        @participant_id,@newsub_group_id,@version_id,
        @assessment,@participant_yend,NULL,
        @fiscal_year,NULL,NULL,@submission_type,@phacode,
        @participant_yend,@phaname,@component_unit_ind,@participant_rpt_to_date,
        @participant_rpt_from_date
        )
        IF @@error <> 0 GOTO SQL_ERROR
```

```
IF UPPER(@intranet) = 'N'
BEGIN
	INSERT INTO participant_assess_action
	(
	object_id,action_code,system_id,
	participant_id,group_id,version_id,assessment_id,
	action_date,user_id,action_amount,action_comment,
	board_resolution_number, certification_date
	)
	VALUES
	(
	1,'CSB','FASPHA',
	@participant_id,@newsub_group_id,@version_id,@assessment,
	getdate(),@user_id,
	NULL,NULL,NULL,NULL
	)
	IF @@error <> 0 GOTO SQL_ERROR
END
ELSE
BEGIN
	INSERT INTO participant_assess_action
	(
	object_id,action_code,system_id,
	participant_id,group_id,version_id,assessment_id,
	action_date,user_id,action_amount,action_comment,
	board_resolution_number,certification_date
	)
	VALUES
	(
	5,'CRM','FASPHA',
	@participant_id,@newsub_group_id,@version_id,@assessment,
	getdate(),@user_id,
	NULL,NULL,NULL,NULL
	)
	IF @@error <> 0 GOTO SQL_ERROR
END


SELECT @assess_action_id=@@identity

	select	@object_id = object_id, @action_code = action_code,
		@system_id = system_id
	from participant_assess_action
	where participant_id = @participant_id
	and group_id = @group_id
```

```
        and version_id = @version_id
        and assessment_id = @assessment
        and participant_assess_action_id = @assess_action_id

IF @@error <> 0 GOTO SQL_ERROR

select @object_assignment_id=MAX(object_assignment_id)
        from object_assignment_queue
        where object_id = @object_id
        and action_code = '@action_code'
        and system_id = @system_id
        and participant_assess_action_id = @assess_action_id
IF @@error <> 0 GOTO SQL_ERROR

if @object_assignment_id = NULL
        select @object_assignment_id=1
else
        select @object_assignment_id=@object_assignment_id + 1

if  UPPER(@intranet) = 'N'
begin
        insert into object_assignment_queue
        (object_id,action_code,system_id,participant_assess_action_id,
        object_assignment_id,
        in_queue_dt,out_queue_dt,userid,assigning_userid)
        values
        (1,'CSB','FASPHA',@assess_action_id,@object_assignment_id,
        getdate(),NULL,@user_id,@user_id)
        IF @@error <> 0 GOTO SQL_ERROR
end
else
begin
        insert into object_assignment_queue
        (object_id,action_code,system_id,participant_assess_action_id,
        object_assignment_id,
        in_queue_dt,out_queue_dt,userid,assigning_userid)
        values
        (5,'CRM','FASPHA',@assess_action_id,@object_assignment_id,
        getdate(),NULL,@user_id,@user_id)
        IF @@error <> 0 GOTO SQL_ERROR
end

        insert into participant_assess_program
        (participant_id,group_id,version_id,assessment_id,program_ref_id,
        accounting_method_ref_id,program_validation_ind,fds_ind)
```

```
        select @participant_id,@newsub_group_id,@version_id,@assessment,
        a.program_ref_id,a.accounting_method_ref_id,'N',a.fds_ind
        from participant_assess_program a
        where a.participant_id=@participant_id
        and a.group_id=@group_id
        and a.version_id=@version_id
        and a.assessment_id=@assessment_id_old
IF @@error <> 0 GOTO SQL_ERROR

insert into participant_assess_pgm_proj
        (participant_id,group_id,version_id,assessment_id,
        program_ref_id,project_id,accounting_method_ref_id,project_validation_ind)
        select @participant_id,@newsub_group_id,@version_id,@assessment,
        b.program_ref_id,b.project_id,b.accounting_method_ref_id,'N'
        from participant_assess_pgm_proj b
        where b.participant_id=@participant_id
        and b.group_id=@group_id
        and b.version_id=@version_id
        and b.assessment_id=@assessment_id_old
IF @@error <> 0 GOTO SQL_ERROR

if (@group_id < @newsub_group_id)
begin
insert into submission_line_item
(participant_id,group_id,version_id,assessment_id,sub_line_item_id,
 program_participant_id,program_group_id,program_version_id,
 program_assessment_id,program_ref_id,item_id,
 item_group_id,item_version_id,threshold_item_id,threshold_group_id,
 threshold_group_version_id,threshold_range_id,sub_line_item_ac_value,
 sub_line_item_ac_text,sub_line_item_ac_smalltext,
 sub_line_item_ac_decimal,sub_line_item_ac_date,accounting_method_ref_id,
 project_id,pgm_proj_participant_id,pgm_proj_group_id,pgm_proj_version_id,
 pgm_proj_assessment_id,pgm_proj_program_ref_id)
 select @participant_id,@newsub_group_id,@version_id,@assessment,
 c.sub_line_item_id,c.program_participant_id,@newsub_group_id,
 c.program_version_id,

 CASE c.program_assessment_id
        WHEN NULL THEN NULL
        ELSE @assessment
 END, c.program_ref_id,

        CASE c.item_id
                WHEN 400166 THEN 400165
                WHEN 400181 THEN 400180
```

```
        WHEN 400183 THEN 400182
        WHEN 400185 THEN 400184
        WHEN 400187 THEN 400186
        WHEN 400208 THEN 400207
        WHEN 400212 THEN 400211
        WHEN 400233 THEN 400232
        WHEN 400238 THEN 400237
        WHEN 400278 THEN 400340
        WHEN 400431 THEN 401008
        WHEN 400428 THEN 400430
        ELSE c.item_id
 END,
        @newsub_group_id,c.item_version_id,

CASE c.threshold_item_id
        WHEN 400166 THEN 400165
        WHEN 400168 THEN 400167
        WHEN 400170 THEN 400169
        WHEN 400261 THEN 400260
        WHEN 400263 THEN 400262
        WHEN 400265 THEN 400264
        WHEN 400267 THEN 400266
        WHEN 400269 THEN 400268
        WHEN 400271 THEN 400270
        WHEN 400173 THEN 400172
        WHEN 400175 THEN 400174
        WHEN 400177 THEN 400176
        WHEN 400179 THEN 400178
        WHEN 400181 THEN 400180
        WHEN 400183 THEN 400182
        WHEN 400185 THEN 400184
        WHEN 400187 THEN 400186
        WHEN 400208 THEN 400207
        WHEN 400212 THEN 400211
        WHEN 400233 THEN 400232
        WHEN 400238 THEN 400237
        WHEN 400278 THEN 400340
        WHEN 400431 THEN 401008
        WHEN 400428 THEN 400430
        ELSE c.threshold_item_id
END,
@newsub_group_id,c.threshold_group_version_id,c.threshold_range_id,
c.sub_line_item_ac_value,c.sub_line_item_ac_text,
c.sub_line_item_ac_smalltext,c.sub_line_item_ac_decimal,
c.sub_line_item_ac_date,c.accounting_method_ref_id,
```

```
        c.project_id,c.pgm_proj_participant_id,c.pgm_proj_group_id,
        c.pgm_proj_version_id,
        CASE c.pgm_proj_assessment_id
                WHEN NULL THEN NULL
                ELSE @assessment
        END,
        c.pgm_proj_program_ref_id
from submission_line_item c
where c.participant_id=@participant_id
  and c.group_id=@group_id
  and c.version_id=@version_id
  and c.assessment_id=@assessment_id_old
  and c.group_id = c.item_group_id
  and ( (@submission_type = 'Noaud' and c.item_id NOT IN (400235,400239,400240))
        OR   (@submission_type <> 'Noaud') )
 and c.item_id NOT IN
(400329,400330,400331,400332,400333,400334,400335,400336,400337,
400338, 401093)
        AND c.item_id NOT BETWEEN 401208 AND 401225

IF @@error <> 0 GOTO SQL_ERROR
end

if (@group_id = @newsub_group_id)
begin
        insert into submission_line_item
        (participant_id,group_id,version_id,assessment_id,sub_line_item_id,
         program_participant_id,program_group_id,program_version_id,
        program_assessment_id,program_ref_id,
         item_id,item_group_id,item_version_id,threshold_item_id,
        threshold_group_id,threshold_group_version_id,threshold_range_id,
        sub_line_item_ac_value,sub_line_item_ac_text,
        sub_line_item_ac_smalltext,sub_line_item_ac_decimal,sub_line_item_ac_date,ac
        counting_method_ref_id,
        project_id,pgm_proj_participant_id,pgm_proj_group_id,
        pgm_proj_version_id,pgm_proj_assessment_id,pgm_proj_program_ref_id)
select @participant_id,@group_id,@version_id,@assessment,c.sub_line_item_id,

c.program_participant_id,c.program_group_id,c.program_version_id,
CASE c.program_assessment_id
WHEN NULL THEN NULL
ELSE @assessment
END,
c.program_ref_id,c.item_id,c.item_group_id,c.item_version_id,
c.threshold_item_id,c.threshold_group_id,c.threshold_group_version_id,
```

```
c.threshold_range_id,  c.sub_line_item_ac_value,c.sub_line_item_ac_text,
c.sub_line_item_ac_smalltext,c.sub_line_item_ac_decimal,
c.sub_line_item_ac_date,c.accounting_method_ref_id,
c.project_id,c.pgm_proj_participant_id,c.pgm_proj_group_id,
c.pgm_proj_version_id,
CASE c.pgm_proj_assessment_id
          WHEN NULL THEN NULL
          ELSE @assessment
          END,
          c.pgm_proj_program_ref_id
          from submission_line_item c
where c.participant_id=@participant_id
        and c.group_id=@group_id
        and c.version_id=@version_id
        and c.assessment_id=@assessment_id_old
       and c.group_id = c.item_group_id
                and c.item_id NOT IN
(400329,400330,400331,400332,400333,400334,400335,400336,400337,
400338,400399,400400,401093)
      and c.item_id NOT BETWEEN 401208 AND 401225

IF @@error <> 0 GOTO SQL_ERROR
end

if (@group_id > @newsub_group_id)
begin
insert into submission_line_item
(participant_id,group_id,version_id,assessment_id,sub_line_item_id,
 program_participant_id,program_group_id,program_version_id,
 program_assessment_id,program_ref_id,
 item_id,item_group_id,item_version_id,threshold_item_id,
 threshold_group_id,threshold_group_version_id,threshold_range_id,
 sub_line_item_ac_value,sub_line_item_ac_text,sub_line_item_ac_smalltext,
 sub_line_item_ac_decimal,sub_line_item_ac_date,accounting_method_ref_id,
 project_id,pgm_proj_participant_id,pgm_proj_group_id,pgm_proj_version_id,
 pgm_proj_assessment_id,pgm_proj_program_ref_id)
 select @participant_id,
          @newsub_group_id,
          @version_id,
          @assessment,
          c.sub_line_item_id,
          c.program_participant_id,
          @newsub_group_id,
          c.program_version_id,
          CASE c.program_assessment_id
```

```
WHEN NULL THEN NULL
ELSE @assessment
END,
c.program_ref_id,
CASE c.item_id
        WHEN 400166 THEN 400165
        WHEN 400167 THEN 400168
        WHEN 400169 THEN 400170
        WHEN 400260 THEN 400261
        WHEN 400262 THEN 400263
        WHEN 400264 THEN 400265
        WHEN 400266 THEN 400267
        WHEN 400268 THEN 400269
        WHEN 400270 THEN 400271
        WHEN 400172 THEN 400173
        WHEN 400174 THEN 400175
        WHEN 400176 THEN 400177
        WHEN 400178 THEN 400179
        WHEN 400180 THEN 400181
        WHEN 400182 THEN 400183
        WHEN 400184 THEN 400185
        WHEN 400186 THEN 400187
        WHEN 400207 THEN 400208
        WHEN 400211 THEN 400212
        WHEN 400232 THEN 400233
        WHEN 400237 THEN 400238
        WHEN 400340 THEN 400278
  ELSE c.item_id
END,


@newsub_group_id,c.item_version_id,

CASE c.threshold_item_id
        WHEN 400166 THEN 400165
        WHEN 400167 THEN 400168
        WHEN 400169 THEN 400170
        WHEN 400260 THEN 400261
        WHEN 400262 THEN 400263
WHEN 400264 THEN 400265
        WHEN 400266 THEN 400267
        WHEN 400268 THEN 400269
        WHEN 400270 THEN 400271
        WHEN 400172 THEN 400173
        WHEN 400174 THEN 400175
```

```
                        WHEN 400176 THEN 400177
                        WHEN 400178 THEN 400179
                        WHEN 400180 THEN 400181
                        WHEN 400182 THEN 400183
                        WHEN 400184 THEN 400185
                        WHEN 400186 THEN 400187
                        WHEN 400207 THEN 400208
                        WHEN 400211 THEN 400212
                        WHEN 400232 THEN 400233
                        WHEN 400237 THEN 400238
                        WHEN 400340 THEN 400278

                        ELSE c.threshold_item_id
                END,

        @newsub_group_id,c.threshold_group_version_id,c.threshold_range_id,
        c.sub_line_item_ac_value,c.sub_line_item_ac_text,
        c.sub_line_item_ac_smalltext,c.sub_line_item_ac_decimal,
        c.sub_line_item_ac_date,c.accounting_method_ref_id,
        c.project_id,c.pgm_proj_participant_id,c.pgm_proj_group_id,
        c.pgm_proj_version_id,
                CASE c.pgm_proj_assessment_id
                        WHEN NULL THEN NULL
                        ELSE @assessment
                END,
        c.pgm_proj_program_ref_id
        from submission_line_item c
        where c.participant_id=@participant_id
                 and c.group_id=@group_id
                 and c.version_id=@version_id
                 and c.assessment_id=@assessment_id_old
                 and c.group_id = c.item_group_id
                                 and c.item_id NOT IN
        400289,400290,400291,400292,400293,400294,400400)
                 and c.item_id NOT IN
        (400329,400330,400331,400332,400333,400334,400335,400336,400337,400338,
        401093)
                 and c.item_id NOT BETWEEN 401208 AND 401225
        IF @@error <> 0 GOTO SQL_ERROR
        end

insert into related_sub_line_item
        (participant_id,group_id,version_id,assessment_id,sub_line_item_id,
        parent_participant_id,parent_group_id,parent_group_version_id,
        parent_assessment_id,parent_sub_line_item_id)
```

```
            select  @participant_id,
                    @newsub_group_id,
                    @version_id,
                    @assessment,
                    d.sub_line_item_id,
                    d.parent_participant_id,
                    @newsub_group_id,
                    d.parent_group_version_id,
                    @assessment,
                    d.parent_sub_line_item_id
            from    related_sub_line_item d
            where d.participant_id=@participant_id
            and d.group_id=@group_id
            and d.version_id=@version_id
            and d.assessment_id=@assessment_id_old

        IF @@error <> 0 GOTO SQL_ERROR
        insert into submission_detail
                (participant_id,group_id,version_id,assessment_id,
                 sub_line_item_id,sub_detail_current_year,sub_detail_current_year_1,
                 sub_detail_current_year_2,sub_detail_current_year_3,
                sub_detail_current_year_4,sub_detail_prior_year)
        select  @participant_id,
                @newsub_group_id,
                @version_id,
                @assessment,
                e.sub_line_item_id,
                e.sub_detail_current_year,
                e.sub_detail_current_year_1,
                e.sub_detail_current_year_2,
                e.sub_detail_current_year_3,
                e.sub_detail_current_year_4,
                e.sub_detail_prior_year
        from submission_detail e
        where e.participant_id=@participant_id
                and e.group_id=@group_id
                and e.version_id=@version_id
                and e.assessment_id=@assessment_id_old

        IF @@error <> 0 GOTO SQL_ERROR

GOTO SQL_ERROR

SQL_ERROR:
```

```
IF @@error > 0
        BEGIN
                SELECT @rc = 12, @reason = 'Failure in SQL Statement'
                SELECT @rc AS RC, @reason AS REASON
                RETURN- 12
        END
ELSE
        BEGIN
                commit tran
                SELECT @rc AS RC, @reason AS REASON
                RETURN 0
        END

GO

IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'reac_faspha_download_sub')
  BEGIN
          PRINT 'CREATE PROCEDURE reac_faspha_download_sub SUCCESSFUL'
          EXEC sp_help reac_faspha_download_sub
  END
ELSE
  BEGIN
           PRINT 'CREATE PROCEDURE reac_faspha_download_sub FAILED'
  END
GO
```

12.0 LOCCS/HUDCAPS INTERFACE CHANGE

12.0 LOCCS/HUDCAPS INTERFACE CHANGE

12.1 Program Description

This is a DCG requirement eliminating the creation of permanent tables within any HUD related databases. A new procedure will be created to insert LOCCS/HUDCAPS data into the REAC database. The sub-routine of creating temporary tables weather it is a database defined permanent or temp tables to house the data before inserting into the REAC database will be eliminated. The new procedure will insert data directly into the database.

New or Modification: Modification

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
	5.4	502	RTM

12.1.1 Software Unit Description

The procedure of creating permanent tables within any HUD related databases would be eliminated. A new procedure will be created to insert LOCCS/HUDCAPS data into the REAC database. The sub-routine of creating temporary tables weather it is a database defined permanent or temp tables to house the data before inserting into the REAC database will be eliminated. The new procedure will insert data directly into the database.

12.1.2 Software Unit

The LOCCS/HUDCAPS interface for uploading data will be modified to allow insertion of data directly into actual tables from temp tables, then automatically deleting the temp tables once data has been loaded into the REAC database.

12.1.3 Accuracy and Validity

Uploaded data into temp tables through the LOCCS/HUDCAPS interface will be loaded into actual tables in the database. Backend database or download of uploaded data will validate that data was inserted into the table.

12.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

12.1.5 Adaptability

When data is loaded into the REACs database, intermediate tables that were created are dropped. This process is inefficient and uses a lot of the database server resources. The new procedure will involve creation of temp tables to hold data coming from LOCCS/HUDCAPS. When the data is loaded into REAC database, the temp tables will be automatically deleted.

12.2 Environment

12.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

12.2.2 Interfaces

The system interfaces with the REAC database, which is used by all the subsystems.

12.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

12.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

12.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

12.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

12.2.3.4 Temporary and Permanent Storage

Temporary tables will be created to hold data until the data is loaded into REAC database tables.

12.2.4 Security

The following table shows user access rights.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

12.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

12.3 Design Details

Screenshot of the LOCCS/HUDCAPS interface.

12-1: LOCCS/HUDCAPS interface

12.3.1 Input

standard_program_set_ind
grants_program_ind
section_eight_program_ind
program_ref_id
program_ref_name
program_ref_cfda_number
program_ref_desc
federal_agency_name
loccs_program_ref_id,
loccs_program_name_desc,

12.3.1.1 Input Records

LOCCS/HUDCAPS data.

12.3.1.2 Input Data Elements

12.3.2 Output

Stored procedure to inspect the upload file count.

12.3.2.1 Output Reports

N/A

12.3.2.2 Output Data Elements

rc
reason
loccs_temp_count
loccs_count

12.3.3 Software Relationships

No changes for Release 8.1.0.0.

12.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
N/A	N/A	N/A	N/A	N/A	N/A	N/A

Software Unit Logic

stored procedures:

faspha_tmp_upload_loccs_records:

```
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_tmp_upload_loccs_record')
  BEGIN
    PRINT 'DROP PROCEDURE faspha_tmp_upload_loccs_record'
    DROP PROCEDURE faspha_tmp_upload_loccs_record
  END
GO

GO

CREATE TABLE temp_loccs_pgm_participant
(participant_id                 numeric(10,0),
 loccs_program_ref_id           varchar(4),
 tin_id                         numeric(10,0) NULL,
 disbursement_amt               numeric(10,0),
 loccs_report_dt                datetime)

GO

CREATE PROCEDURE faspha_tmp_upload_loccs_record
@participant_id                 numeric(10,0),
@loccs_program_ref_id           varchar(4),
@tin_id                         numeric(10,0),
@disbursement_amt               numeric(10,0),
@loccs_report_dt                datetime

AS

IF @@TRANCOUNT = 0
        SET CHAINED OFF

DECLARE     @rc                             integer,
            @reason                         varchar(255),
            @returnErr                      integer,
            @loccs_program_ref_id_count     integer

SELECT @rc = 0, @reason = NULL, @loccs_program_ref_id_count = 0
```

```
INSERT INTO temp_loccs_pgm_participant
(participant_id,loccs_program_ref_id,tin_id,disbursement_amt,loccs_report_dt)
VALUES (@participant_id,@loccs_program_ref_id,
        @tin_id,@disbursement_amt,
        @loccs_report_dt)
IF @@error <> 0
        BEGIN
                SELECT @rc=1, @reason='Could not insert into temp table'
                GOTO SQL_ERROR
        END


SELECT @loccs_program_ref_id_count = COUNT(*)
        FROM loccs_program_ref
        WHERE loccs_program_ref_id = @loccs_program_ref_id
IF @@error <> 0
        BEGIN
                SELECT @rc=2, @reason='Error selecting count of records in loccs_program_ref
table'
                GOTO SQL_ERROR
        END


IF @loccs_program_ref_id_count = 0
        BEGIN
                SELECT @rc=3, @reason='Record does not exist in loccs_program_ref table'
                GOTO SQL_ERROR
        END


GOTO SQL_ERROR

SQL_ERROR:

IF @@error > 0
        BEGIN
                SELECT @rc = 12, @reason = 'Failure in SQL Statement'
                SELECT @rc AS RC, @reason AS REASON
                RETURN- 12
        END
ELSE
        BEGIN
                SELECT @rc AS RC, @reason AS REASON
                RETURN 0
        END
```

```
GO
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_tmp_upload_loccs_record')
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_tmp_upload_loccs_record  SUCCESSFUL'
    EXEC sp_help faspha_tmp_upload_loccs_record
  END
ELSE
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_tmp_upload_loccs_record FAILED'
  END
GO
```

faspha_upload_loccs_record:

```
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_upload_loccs_record')
  BEGIN
    PRINT 'DROP PROCEDURE faspha_upload_loccs_record'
    DROP PROCEDURE faspha_upload_loccs_record
  END
GO

CREATE PROCEDURE faspha_upload_loccs_record

@loccs_report_dt                datetime

table

AS

IF @@TRANCOUNT = 0
        SET CHAINED OFF

DECLARE   @rc                   integer,
          @reason               varchar(255),
          @loccs_temp_count     integer,
          @loccs_count          integer

SELECT @rc = 0, @reason = NULL, @loccs_temp_count = 0, @loccs_count = 0

SELECT    @loccs_temp_count = COUNT(*)
FROM      temp_loccs_pgm_participant
WHERE     loccs_report_dt = @loccs_report_dt
IF @@error <> 0
```

```
        BEGIN
                SELECT @rc=1, @reason='Error counting total records from temp table'
                GOTO SQL_ERROR
        END

SELECT @loccs_count = COUNT(*)
        FROM loccs_pgm_participant_report
        WHERE loccs_report_dt = @loccs_report_dt
IF @@error <> 0
        BEGIN
                SELECT @rc=2, @reason='Error counting total records from
                loccs_pgm_participant_report table'
                GOTO SQL_ERROR
        END

IF @loccs_count > 0
        BEGIN
                SELECT @rc=3, @reason='Records exist for the upload period'
                GOTO SQL_ERROR
        END
ELSE

IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_upload_loccs_record')
  BEGIN
    PRINT 'DROP PROCEDURE faspha_upload_loccs_record'
    DROP PROCEDURE faspha_upload_loccs_record
  END
GO


CREATE PROCEDURE faspha_upload_loccs_record

@loccs_report_dt              datetime

table

AS

IF @@TRANCOUNT = 0
        SET CHAINED OFF

DECLARE   @rc                         integer,
          @reason                     varchar(255),
          @loccs_temp_count           integer,
```

```
        @loccs_count            integer

SELECT @rc = 0, @reason = NULL, @loccs_temp_count = 0, @loccs_count = 0

-- Get record count from temp table
SELECT  @loccs_temp_count = COUNT(*)
    FROM temp_loccs_pgm_participant
    WHERE   loccs_report_dt = @loccs_report_dt
IF @@error <> 0
    BEGIN
        SELECT @rc=1, @reason='Error counting total records from temp table'
        GOTO SQL_ERROR
    END

-- Get period record count of table prior to insertion
SELECT @loccs_count = COUNT(*)
    FROM loccs_pgm_participant_report
    WHERE loccs_report_dt = @loccs_report_dt
IF @@error <> 0
    BEGIN
        SELECT @rc=2, @reason='Error counting total records from
loccs_pgm_participant_report table'
        GOTO SQL_ERROR
    END

-- Verify if records exist for upload period
IF @loccs_count > 0
    BEGIN
        SELECT @rc=3, @reason='Records exist for the upload period'
        GOTO SQL_ERROR
    END
ELSE
    -- Insert records into loccs_pgm_participant_report table
    BEGIN
        BEGIN TRANSACTION
            INSERT INTO loccs_pgm_participant_report (participant_id,
loccs_program_ref_id, tin_id, disbursement_amt, loccs_report_dt)
            SELECT a.participant_id, a.loccs_program_ref_id, a.tin_id,
a.disbursement_amt, a.loccs_report_dt
            FROM temp_loccs_pgm_participant a NOHOLDLOCK,
pih_all_participant b NOHOLDLOCK, loccs_program_ref c NOHOLDLOCK
            WHERE a.loccs_program_ref_id = c.loccs_program_ref_id
            AND a.participant_id = b.participant_id
            AND a.loccs_report_dt = @loccs_report_dt
            IF @@error <> 0
```

```
                        BEGIN
                                SELECT @rc=4, @reason='Error inserting records into
                                #temp_loccs_pgm_participant table'
                                GOTO SQL_ERROR
                        END
                END

                SELECT @loccs_count = COUNT(*)
                FROM loccs_pgm_participant_report
                WHERE loccs_report_dt = @loccs_report_dt
        IF @@error <> 0
                BEGIN
                        SELECT @rc=5, @reason='Error counting total records from
loccs_pgm_participant_report table'
                        GOTO SQL_ERROR
                END

        IF @loccs_temp_count <> @loccs_count
                BEGIN
                        DELETE FROM loccs_pgm_participant_report
                        WHERE loccs_report_dt = @loccs_report_dt
                        IF @@error <> 0
                                BEGIN
                                        SELECT @rc=6, @reason='Error deletion of loccs records
                                        from loccs_pgm_participant_report table'
                                        GOTO SQL_ERROR
                                END
                        SELECT @rc=7, @reason='Table record count does not equal temp
                        record count'
                END
                IF @@error <> 0
                        BEGIN
                                SELECT @rc=8, @reason='Error checking the temp record count
                                with the table record count'
                                GOTO SQL_ERROR
                        END

GOTO SQL_ERROR

SQL_ERROR:

IF @@error > 0
        BEGIN
                SELECT @rc = 12, @reason = 'Failure in SQL Statement'
                SELECT 'ROLLBACK TRANSACTION', @rc AS RC, @reason AS REASON
```

```
                ROLLBACK TRANSACTION
                RETURN- 12
        END
ELSE
        BEGIN
                SELECT @rc AS RC, @reason AS REASON, @loccs_count AS loccs_count
                COMMIT TRANSACTION
                RETURN 0
        END

GO
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_upload_loccs_record')
  BEGIN
         PRINT 'CREATE PROCEDURE faspha_upload_loccs_record  SUCCESSFUL'
         EXEC sp_help faspha_upload_loccs_record
  END
ELSE
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_upload_loccs_record FAILED'
  END
GO
                    BEGIN
                BEGIN TRANSACTION
                        INSERT INTO loccs_pgm_participant_report (participant_id,
                        loccs_program_ref_id, tin_id,
                        disbursement_amt, loccs_report_dt)
                        SELECT a.participant_id, a.loccs_program_ref_id, a.tin_id,
                        a.disbursement_amt, a.loccs_report_dt
                        FROM temp_loccs_pgm_participant a NOHOLDLOCK,
                        pih_all_participant b NOHOLDLOCK,
                        loccs_program_ref c NOHOLDLOCK
                        WHERE a.loccs_program_ref_id = c.loccs_program_ref_id
                        AND a.participant_id = b.participant_id
                        AND a.loccs_report_dt = @loccs_report_dt
                        IF @@error <> 0
                                BEGIN
                                        SELECT @rc=4, @reason='Error inserting records into
                                        #temp_loccs_pgm_participant table'
                                        GOTO SQL_ERROR
                                END
                END

                SELECT @loccs_count = COUNT(*)
                FROM loccs_pgm_participant_report
```

```
            WHERE loccs_report_dt = @loccs_report_dt
        IF @@error <> 0
            BEGIN
                SELECT @rc=5, @reason='Error counting total records from
                loccs_pgm_participant_report table'
                GOTO SQL_ERROR
            END

            IF @loccs_temp_count <> @loccs_count
            BEGIN
                DELETE FROM loccs_pgm_participant_report
                WHERE loccs_report_dt = @loccs_report_dt
                IF @@error <> 0
                    BEGIN
                        SELECT @rc=6, @reason='Error deletion of loccs records
                        from
                        loccs_pgm_participant_report table'
                        GOTO SQL_ERROR
                    END
                SELECT @rc=7, @reason='Table record count
                    does not equal temp record count'
            END
            IF @@error <> 0
                BEGIN
                    SELECT @rc=8, @reason='Error checking the
                        temp record count with the table record count'
                    GOTO SQL_ERROR
                END

GOTO SQL_ERROR

SQL_ERROR:

IF @@error > 0
        BEGIN
            SELECT @rc = 12, @reason = 'Failure in SQL Statement'
            SELECT 'ROLLBACK TRANSACTION', @rc AS RC, @reason AS REASON
            ROLLBACK TRANSACTION
            RETURN- 12
        END
ELSE
        BEGIN
            SELECT @rc AS RC, @reason AS REASON, @loccs_count AS loccs_count
            COMMIT TRANSACTION
            RETURN 0
```

```
        END

GO
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_upload_loccs_record')
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_upload_loccs_record  SUCCESSFUL'
    EXEC sp_help faspha_upload_loccs_record
  END
ELSE
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_upload_loccs_record FAILED'
  END
GO
```

faspha_tmp_upload_loccs_pgm:

```
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_tmp_upload_loccs_pgm')
  BEGIN
    PRINT 'DROP PROCEDURE faspha_tmp_upload_loccs_pgm'
    DROP PROCEDURE faspha_tmp_upload_loccs_pgm
  END
GO

CREATE TABLE temp_loccs_pgm
(loccs_program_ref_id         varchar(4),
 loccs_program_name_desc      char(255),
 program_ref_id               char(20))

GO

CREATE PROCEDURE faspha_tmp_upload_loccs_pgm

@loccs_program_ref_id             varchar(4),
@loccs_program_name_desc          char(255),
@program_ref_id                   char(20)

AS

IF @@TRANCOUNT = 0
        SET CHAINED OFF

DECLARE     @rc                       integer,
            @reason                   varchar(255),
```

```
        @returnErr              integer,
        @program_ref_count      integer

SELECT @rc = 0, @reason = NULL, @program_ref_count = 0


INSERT INTO temp_loccs_pgm
(loccs_program_ref_id,loccs_program_name_desc,program_ref_id)
VALUES (@loccs_program_ref_id,
        @loccs_program_name_desc,
        @program_ref_id)
IF @@error <> 0
        BEGIN
                SELECT @rc=1, @reason='Could not insert into temp table'
                GOTO SQL_ERROR
        END


SELECT @program_ref_count = COUNT(*)
        FROM program_ref
        WHERE program_ref_id = @program_ref_id
IF @@error <> 0
        BEGIN
                SELECT @rc=2, @reason='Error selecting count of records in program_ref table'
                GOTO SQL_ERROR
        END


IF @program_ref_count = 0
        BEGIN
                SELECT @rc=3, @reason='Record does not exist in program_ref table'
                GOTO SQL_ERROR
        END


GOTO SQL_ERROR

SQL_ERROR:

IF @@error > 0
        BEGIN
                SELECT @rc = 12, @reason = 'Failure in SQL Statement'
                SELECT @rc AS RC, @reason AS REASON
                RETURN- 12
        END
```

```
ELSE
        BEGIN
                SELECT @rc AS RC, @reason AS REASON
                RETURN 0
        END


GO
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_tmp_upload_loccs_pgm')
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_tmp_upload_loccs_pgm  SUCCESSFUL'
    EXEC sp_help faspha_tmp_upload_loccs_pgm
  END
ELSE
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_tmp_upload_loccs_pgm FAILED'
  END
GO
```

faspha_upload_loccs_pgm:

```
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_upload_loccs_pgm')
  BEGIN
    PRINT 'DROP PROCEDURE faspha_upload_loccs_pgm'
    DROP PROCEDURE faspha_upload_loccs_pgm
  END
GO

CREATE PROCEDURE faspha_upload_loccs_pgm

AS

IF @@TRANCOUNT = 0
        SET CHAINED OFF

DECLARE   @rc                          integer,
          @reason                      varchar(255),
          @loccs_pgm_temp_count        integer,
          @loccs_start_pgm_count       integer,
          @loccs_end_pgm_count         integer

SELECT @rc = 0, @reason = NULL, @loccs_pgm_temp_count = 0, @loccs_start_pgm_count =
0, @loccs_end_pgm_count = 0
```

```
SELECT @loccs_start_pgm_count = COUNT(*)
        FROM loccs_program_ref NOHOLDLOCK
IF @@error <> 0
        BEGIN
                SELECT @rc=1, @reason='Error getting total count from loccs_program_ref
                table'
                GOTO SQL_ERROR
        END

SELECT      @loccs_pgm_temp_count = COUNT(*)
FROM        emp_loccs_pgm
IF @@error <> 0
        BEGIN
                SELECT @rc=2, @reason='Error counting total records from temp table'
                GOTO SQL_ERROR
        END

BEGIN
        BEGIN TRANSACTION
                INSERT INTO loccs_program_ref
                (loccs_program_ref_id,loccs_program_name_desc,
                program_ref_id)
                SELECT loccs_program_ref_id, loccs_program_name_desc, program_ref_id
                FROM temp_loccs_pgm NOHOLDLOCK
                IF @@error <> 0
                        BEGIN
                                SELECT @rc=3, @reason='Error inserting records into
                                temp_loccs_pgm table'
                                GOTO SQL_ERROR
                        END
        END

SELECT @loccs_end_pgm_count = COUNT(*)
        FROM loccs_program_ref NOHOLDLOCK
IF @@error <> 0
        BEGIN
                SELECT @rc=4, @reason='Error counting total records from loccs_program_ref
                table'
                GOTO SQL_ERROR
        END

IF @loccs_start_pgm_count + @loccs_pgm_temp_count <> @loccs_end_pgm_count
        BEGIN
                SELECT @rc = 5, @reason = 'Error - temp record count does not match table
                record count'
```

```
        END
        IF @@error <> 0
                BEGIN
                        SELECT @rc=6, @reason='Error checking the temp record count with the
                        table record count'
                        GOTO SQL_ERROR
                END

GOTO SQL_ERROR

SQL_ERROR:

IF (@@error > 0 OR @rc > 0)
        BEGIN
                SELECT 'ROLLBACK TRANSACTION', @rc AS RC, @reason AS REASON
                ROLLBACK TRANSACTION
                RETURN- 12
        END
ELSE
        BEGIN
                SELECT @rc AS RC, @reason AS REASON
                COMMIT TRANSACTION
                RETURN 0
        END

GO
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_upload_loccs_pgm')
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_upload_loccs_pgm  SUCCESSFUL'
    EXEC sp_help faspha_upload_loccs_pgm
  END
ELSE
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_upload_loccs_pgm FAILED'
  END
GO
```

faspha_tmp_upload_pgm_ref:

```
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_tmp_upload_pgm_ref')
  BEGIN
    PRINT 'DROP PROCEDURE faspha_tmp_upload_pgm_ref'
    DROP PROCEDURE faspha_tmp_upload_pgm_ref
```

```
  END
GO

GO

CREATE TABLE temp_pgm_ref
(standard_program_set_ind    char(1),
 grants_program_ind          char(1),
 section_eight_program_ind   char(1),
 program_ref_id              varchar(20),
 program_ref_name            varchar(70),
 program_ref_cfda_number     varchar(20),
 program_ref_desc            varchar(255),
 federal_agency_name         varchar(100))

GO

CREATE PROCEDURE faspha_tmp_upload_pgm_ref

@standard_program_set_ind       char(1),
@grants_program_ind             char(1),
@section_eight_program_ind      char(1),
@program_ref_id                 varchar(20),
@program_ref_name               varchar(70),
@program_ref_cfda_number        varchar(20),
@program_ref_desc               varchar(255),
@federal_agency_name            varchar(100)

AS

IF @@TRANCOUNT = 0
        SET CHAINED OFF

DECLARE   @rc                   integer,
          @reason               varchar(255),
          @returnErr            integer

SELECT    @rc = 0,
          @reason = NULL

INSERT INTO temp_pgm_ref
(standard_program_set_ind,grants_program_ind,section_eight_program_ind,
 program_ref_id,
 program_ref_name,program_ref_cfda_number,program_ref_desc,federal_agency_name)
VALUES    (@standard_program_set_ind,
```

```
                @grants_program_ind,
                @section_eight_program_ind,
                @program_ref_id,
                @program_ref_name,
                @program_ref_cfda_number,
                @program_ref_desc,
                @federal_agency_name
                )
IF @@error <> 0
        BEGIN
                SELECT @rc=1, @reason='Could not insert into temp table'
                GOTO SQL_ERROR
        END

GOTO SQL_ERROR

SQL_ERROR:

IF @@error > 0
        BEGIN
                SELECT @rc = 12, @reason = 'Failure in SQL Statement'
                SELECT @rc AS RC, @reason AS REASON
                RETURN- 12
        END
ELSE
        BEGIN
                SELECT @rc AS RC, @reason AS REASON
                RETURN 0
        END

GO
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name =
'faspha_tmp_upload_pgm_ref')
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_tmp_upload_pgm_ref  SUCCESSFUL'
    EXEC sp_help faspha_tmp_upload_pgm_ref
  END
ELSE
  BEGIN
    PRINT 'CREATE PROCEDURE faspha_tmp_upload_pgm_ref FAILED'
  END
GO
```

faspha_upload_pgm_ref:

```
IF EXISTS (SELECT *
FROM sysobjects WHERE type = 'P' AND name = 'faspha_upload_pgm_ref')
  BEGIN
    PRINT 'DROP PROCEDURE faspha_upload_pgm_ref'
    DROP PROCEDURE faspha_upload_pgm_ref
  END
GO


CREATE PROCEDURE faspha_upload_pgm_ref

AS

IF @@TRANCOUNT = 0
        SET CHAINED OFF

DECLARE     @rc                     integer,
            @reason                 varchar(255),
            @pgm_temp_count         integer,
            @start_pgm_count        integer,
            @end_pgm_count          integer

SELECT      @rc = 0,
            @reason = NULL,
            @pgm_temp_count = 0,
            @start_pgm_count = 0,
            @end_pgm_count = 0

SELECT @start_pgm_count = COUNT(*)
FROM program_ref NOHOLDLOCK

IF @@error <> 0
        BEGIN
                SELECT @rc=1, @reason='Error getting total count from program_ref table'
                GOTO SQL_ERROR
        END

SELECT      @pgm_temp_count = COUNT(*)
FROM        temp_pgm_ref
IF @@error <> 0
        BEGIN
                SELECT @rc=2, @reason='Error counting total records from temp table'
                GOTO SQL_ERROR
        END
```

```
BEGIN
        BEGIN TRANSACTION
                INSERT INTO program_ref
                (program_ref_id,program_ref_name,program_ref_cfda_number,
                program_ref_desc,
                federal_agency_name,standard_program_set_ind,grants_program_ind,
                section_eight_program_ind)
                SELECT program_ref_id,program_ref_name,program_ref_cfda_number,
                        program_ref_desc,federal_agency_name,
                        standard_program_set_ind,grants_program_ind,
                        section_eight_program_ind
                FROM   temp_pgm_ref NOHOLDLOCK
                IF @@error <> 0
                        BEGIN
                                SELECT @rc=3, @reason='Error inserting records into
                                program_ref table'
                                GOTO SQL_ERROR
                        END
        END

SELECT @end_pgm_count = COUNT(*)
FROM    program_ref NOHOLDLOCK
IF @@error <> 0
        BEGIN
                SELECT @rc=4, @reason='Error counting total records from program_ref table'
                GOTO SQL_ERROR
        END

IF @start_pgm_count + @pgm_temp_count <> @end_pgm_count
        BEGIN
                SELECT @rc = 5, @reason = 'Error - temp record count does not match table
                record count'
        END
        IF @@error <> 0
                BEGIN
                        SELECT @rc=6, @reason='Error checking the temp record count with the
                        table record count'
                        GOTO SQL_ERROR
                END

GOTO SQL_ERROR

SQL_ERROR:

IF (@@error > 0 OR @rc > 0)
```

```
        BEGIN
                SELECT 'ROLLBACK TRANSACTION', @rc AS RC, @reason AS REASON
                ROLLBACK TRANSACTION
                RETURN- 12
        END
ELSE
        BEGIN
                SELECT @rc AS RC, @reason AS REASON
                COMMIT TRANSACTION
                RETURN 0
        END

GO
IF EXISTS (SELECT *
              FROM sysobjects
              WHERE type = 'P'
                   AND name = 'faspha_upload_pgm_ref')
  BEGIN
     PRINT 'CREATE PROCEDURE faspha_upload_pgm_ref  SUCCESSFUL'
     EXEC sp_help faspha_upload_pgm_ref
  END
ELSE
  BEGIN
     PRINT 'CREATE PROCEDURE faspha_upload_pgm_ref FAILED'
  END
GO
```

13.0 Review Submission Page Text Change

13.0 REVIEW SUBMISSION PAGE TEXT CHANGE

13.1 Program Description

On the "Review Submission" page under the "Financial Statement" tab of the "Notes and Findings" page, modification will be made to the text on the description column. The text currently reads: "Government-Wide or General purpose Financial Statement". The text will be changed to read: "Government-Wide and or Fund Financial Statement".

New or Modification: Modification

Screen Traceability: The following table traces the screen design to one or more business rules, process numbers, functional requirements, and issue tracking references (if applicable).

Business Rule #	Process #	Functional Requirement #	Source
1.1-1, 2.1-1	2		RTM

13.1.1 Software Unit Description

On the "Review Submission" page under the "Financial Statement" tab of the "Notes and Findings" page, modification will be made to the text on the description column. The text currently reads: "Government-Wide or General purpose Financial Statement". The text will be changed to read: "Government-Wide and or Fund Financial Statement".

13.1.2 Software Unit

On the "Notes and Findings" page of the Financial Data Schedule section of the internal and external FASS-PH, "Financial Statement" tab line item G5150-010 description column text will be changed to read: "Government-Wide and or Fund Financial Statement".

13.1.3 Accuracy and Validity

Text change for line item G5150-010 description column text will be displayed on the screen.

13.1.4 Timing

FASS-PH must conform to the REAC standards of timing and timely operation. The following is a more detailed table of the timing information expected by the program within FASS-PH:

PERFORMANCE MEASURE	TESTING PROCEDURE
Timing	
FASS-PH will score submitted PHA submissions every evening.	After the programs have run in the REAC Nightly Batch in test environments, the database will be queried to confirm that the scoring procedures have executed properly throughout the testing phase.
FASS-PH will be available for PHA submission entry 24 hours per day, except for downtime during scheduled system maintenance and upgrades.	Multiple submissions will be made throughout the duration of the testing phase.
FASS-PH will provide capability to generate multiple reports 24 hours per day except for downtime during scheduled system maintenance and upgrades.	Multiple reports will be generated throughout the duration of the testing phase.
FASS-PH will provide capability to load every page in 8 seconds or less (except for waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.
FASS-PH will provide capability to execute every stored procedure in 3 seconds or less (except for stored procedures on waived pages).	Timing requirements cannot be fully tested. REAC does not have an automated testing tool capable of testing these requirements. However, the testing team will execute tests scripts with the debug application turned on in order to monitor load time in system test.

13.1.5 Adaptability

Flexibility of FASS-PH will be enhanced by this requirement and text changes are needed to adhere to accounting principles.

13.2 Environment

13.2.1 Support Software Environment

The support software environment is comprised of the following software packages:

Software	Description
Sybase	A widely used database management tool.
HTML	A widely used and accepted scripting language.
JavaScript	A widely used and accepted scripting language.
Cold Fusion Studio 5.0	A widely used application development tool.
ErWin	This tool allows the development team to develop the logical and physical models for the project.
PVCS Version Manager	This tool is used for configuration management.
PVCS Tracker	This tool is used for reporting and tracking system change requests.
SQL Advantage	This tool serves as an interface to the Sybase database.
SQL Programmer	This tool serves as an interface to the Sybase database.
Netscape 4.79, Internet Explorer 6.0 SP2	Browsers that serve as the default web browsers for the FASS-PH application.
Adobe Acrobat Reader Version 5.0 or higher	This tool is used to create the Data Collection Form Report.
XML Tools	These tools are used for developing and managing XML assets.
Dream Weaver	This tool is used for developing and managing web sites.

13.2.2 Interfaces

The Financial Statement section of the “Notes and Findings” page does not interface with other systems.

13.2.3 Storage

This capability does not require significant additional space. DCG will determine if additional space is necessary. For more information, please refer to Section 3.1.2 (Storage Medium) in the

System/Subsystem Specifications document. Refer to HUD Handbook 2229.1 Records Disposition Scheduling for Automated System for further information on data storage.

13.2.3.1 Internal Storage

There are no new internal storage requirements to support this capability.

13.2.3.2 Device Storage

There are no new device storage requirements to support this capability.

13.2.3.3 Offline Storage

There are no new offline storage requirements to support this capability.

13.2.3.4 Temporary and Permanent Storage

There are no new temporary and permanent storage requirements to support this capability.

13.2.4 Security

The following table shows user access rights for the DCF/Financial Statement reports.

Role	Create	Read	Update	Delete	Detailed Description
PID		☒			Users can only view the information.
FIA		☒			Users can only view the information.
SMT		☒			Users can only view the information.
CPV		☒			Users can only view the information.
FID		☒			Users can only view the information.
AM		☒			Users can only view the information.
RFA		☒			Users can only view the information.
GUS		☒			Users can only view the information.
MRA		☒			Users can only view the information.

13.2.5 Communications Environment

FASS-PH only requires that users access the system via the Internet or Intranet. All communications are done over this medium.

13.4 Design Details

The following screenshot of the "Financial Statements" tab description column text.

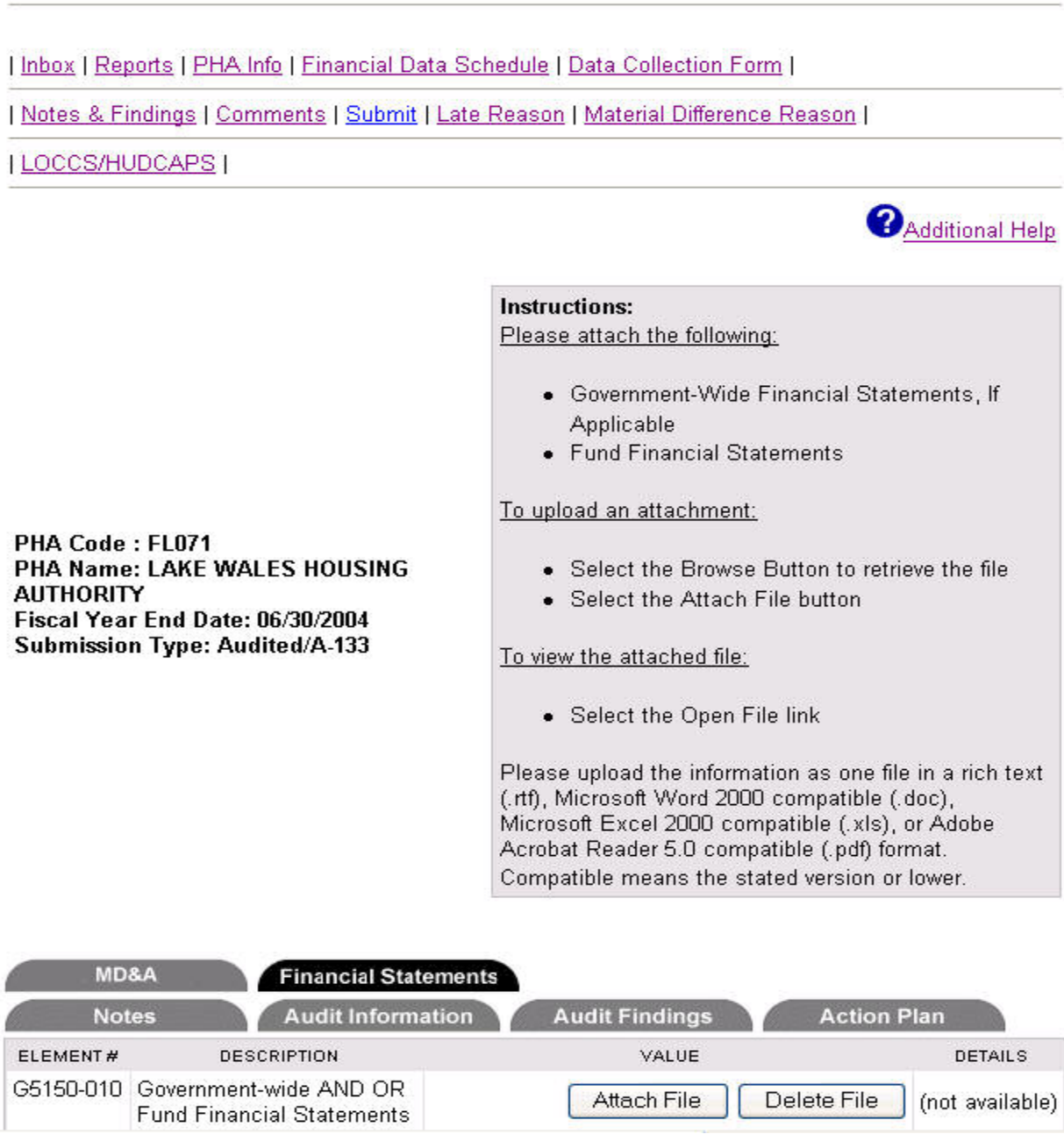


Figure 13-1: NOTES AND FINDINGS /FINANCIAL STATEMENT PAGE

13.3.1 Input

No input records for this requirement

13.3.1.1 Input Records

The line items on the database are used as input to the process. There are no actual input records.

13.3.1.2 Input Data Elements

The line items on the database are used as input to the process.

13.3.2 Output

The text changes are shown on the screen.

13.3.2.1 Output Reports

The text changes are shown on the screen.

13.3.2.2 Output Data Element

13.3.3 Software Relationships

No changes for Release 8.1.0.0

13.3.4 Software Unit Logic

The following sections contain the detailed software unit logic.

Object Action Table

Field Name	Editable by REAC (Source System)	Datastore/ Reference Table	Data Element	Data Element Type/ Length	Values	Description/Comments
N/A	N/A	N/A	N/A	N/A	N/A	N/A

Software Unit Logic

On the "Notes and Findings" page of the Financial Data Schedule section of the internal and external FASS-PH, "Financial Statement" tab line item G5150-010 description column text will be changed to read: "Government-Wide and or Fund Financial Statement".

Stored Procedures

N/A

APPENDIX A: BUSINESS RULES

Business Process	Req. #	Functional Requirement	Source	S	I	U	Mod/Dup	Release Identified
1		**Complete and Submit Financial Information**						
1.1		**Access System**						
	497	WASS will remove the guest checkbox on the Login interface. Have the system automatically recognize a guest user.	User Requirement					8.1.0.0
	505	Remove any instances of http port in ColdFusion templates and replace http port with the relative server.	User Requirement					8.1.0.0
1.2		**Download Submission Template**						
1.3		**Download Draft Submission Data**						
1.4		**Input Data**						
	501	Change the storage of permanent file attachments from being part of the UNIX /Windows file system to being stored as Binary Large Objects (BLOBs) in the database. All file attachments need to be stored and retrieved on the REACS database.	User Requirement					8.1.0.0
	503	Remove identity attribute from the column definition in the assessment table and replace the attribute with a stored procedure to find the sequential primary key value.	User Requirement					8.1.0.0
1.4.1		**Input FDS**						
1.4.2		**Input Data Collection Form**						
	494	First, the DCF/Financial Statement/G3000-010 Type of Audit Report/G3000-060 & 070 will now reflect Fund Type and Opinion of the Fund rather than Program. Auditors should only be entering opinion for funds within the PHA.	User Requirement					8.1.0.0
	495	Modify Line Item 4200-050 to default to "N/A"; if and only if Line Item 4200-010 is selected "No" for Non-Major Programs audited A133, there will be no penalty when this opinion is selected.	User Requirement					8.1.0.0
	496	New logic will be in place, so that the external user will not be able to enter any amount on Line Item G1102. This new methodology should begin for all 9/30/2005 submissions.	User Requirement					8.1.0.0
1.4.3		**Input Notes and Findings**						
1.4.4		**Input Material Difference Reason**						
1.5		**Validate Data Format**						
1.6		**Save Draft Financial Data**						
1.6.1		**Transmission Error Handling (rollback)**						
1.6.2		**Post Successful Submission Status**						

Business Process	Req. #	Functional Requirement	Source	S	I	U	Mod/Dup	Release Identified
1.7		**Validate Data Against Business Rules**						
1.7.1		**Generate Data Error Report**						
1.7.2		**Correct Data Issues**						
1.8		**Validate Data Against Edit Flags**						
1.8.1		**Run External Program-Based Edit Flags**						
1.8.2		**Run External Entity-Wide Edit Flags**						
1.9		**Submit Final Submission Data**						
1.9.1		**Submit for IPA Review**						
1.9.1.1		**Perform IPA Review**						
1.9.1.2		**Submit to PHA or Section 8 Entity for Corrections**						
1.9.2		**Submit to REAC via Online System**						
1.9.2.1		**Transmission Error Handling (rollback)**						
1.9.2.2		**Post Successful Submission Status**						
1.9.3		**Submit to REAC via XML Interface**						
1.9.3.1		**Transmission Error Handling (rollback)**						
1.9.3.2		**Post Successful Submission Status**						
2		**Receive Financial Information**						
2.1		**Post Financial Information**						
2.2		**Display Financial Information for HUD User**						
	504	Allow analysts to review prior year submission comments while still reviewing the current FYE submission. This will allow the analyst to review prior submission comments without navigating between multiple submissions.	User Requirement					8.1.0.0
2.3		**Display Financial Information for an External User**						
	498	Modify the FASS Analyst column for the external user inbox only to display the name of the Business Manager or Analyst.	User Requirement					8.1.0.0
	505	Display Financial Information for an External User	User Requirement					
2.4		**Calculate Score**						
2.4.1		**Calculate Current Ratio**						
2.4.2		**Calculate Number of Months Expendable Fund Balance**						
2.4.3		**Calculate Tenant Receivable Outstanding**						
2.4.4		**Calculate Occupancy Loss**						

Business Process	Req. #	Functional Requirement	Source	S	I	U	Mod/Dup	Release Identified
2.4.5		Calculate Expense Management/Utility Consumption						
2.4.6		Calculate Net Income or Loss Divided by the Expendable Fund Balance						
2.4.7		Assess Penalty Points for Audit Flags						
2.4.8		Assess Penalty Points for Materiality						
2.5		Assign Risk Level						
2.6		Run Internal Edit Flags						
2.7		Run Auto Acceptance Process						
3		Assess Financial Condition						
3.1		Review Submission						
3.1.1		Review IPA Comments from FASS-QA						
3.1.2		Compare FDS Unit Count to REAC Database Unit Count						
3.2		Adjust Score						
3.2.1		Abate Penalty Points for Audit Flags						
3.2.2		Abate Penalty Points for Materiality						
3.2.3		Abate Penalty Points for Late Submission						
3.3		Flag Submission for FASS-QA						
3.4		Record Comments						
3.5		Accept/Reject Submission						
3.5.1		Provide Data for Rejection/Resubmission Notification						
3.5.2		Provide data for conditional acceptance notification						
3.6		Submit Review						
3.7		Invalidate Submission						
3.8		Transfer Accepted DCF Data to the OMB (Audited/A-133 Submissions Only)						
4		Generate System Reports						
4.1		Generate FDS Report (Current Year – Prior Year)						
	499	Repair the FDS report page to print correctly from MS Internet Explorer.	User Requirement					8.1.0.0
4.2		Generate FDS Report (Audited – Unaudited)						
4.3		Generate Data Collection Form Report						
4.4		Generate Combined Balance Sheet						
4.5		Generate Combined Statement of Revenues and Expenses						

Business Process	Req. #	Functional Requirement	Source	S	I	U	Mod/Dup	Release Identified
4.6		**Generate Schedule of Federal Financial Assistance**						
4.7		**Generate Financial Statement Footnotes**						
4.8		**Generate REAC Management Reports**						
4.8.1		**Generate Submission Summary**						
4.8.2		**Generate Pending Submission by Reviewer**						
4.8.3		**Generate Pending Submission by Date**						
4.8.4		**Generate Production Report**						
4.8.5		**Generate Individual Output Report**						
4.8.6		**Generate High Reserve and Liquidity Adjustment Report**						
4.8.7		**Generate Pending Review Aging Schedule Report**						
4.8.8		**Generate PHA History Report**						
4.9		**Generate IPA Status Reports**						
4.10		**Generate Statement of Net Assets**						
4.11		**Generate Statement of Revenues, Expenses, and Changes in Fund Net Assets**						
4.12		**Generate LOCCS Comparison Report and HUDCAPS**						
4.13		**Generate Edit Flags Report**						
5		**Manage Thresholds**						
5.1		**Override Risk Level**						
5.2		**Help Maintenance**						
5.3		**Maintain Auto Acceptance Criteria**						
5.4		**Maintain LOCCS/HUDCAPS Data**						
	502	Remove storing Line of Credit Control System/ HUD Central Accounting Processing System (LOCCS/HUDCAPS) data in permanent tables and pipe the HOCCS/HUDCAPS data directly into the REAC database.						8.1.0.0
5.5		**Workload Management**						
5.6		**OMB Compliance Statement**						